BASIC
BAJNOKSÁG

Szki

Készült a GRIFF Propaganda G.M.
T.b. Rákóczi Nyomda közreműködésével
Munkaszám: 84619

ISBN – 963 – 016080 3

SZÁMÍTÁSTECHNIKAI
KOORDINÁCIÓS INTÉZET

BASIC BAJNOKSÁG

BUDAPEST, 1984

© SZÁMÍTÁSTECHNIKAI KOORDINÁCIÓS INTÉZET, 1984

Jelen kiadvány az SZKI előzetes írásbeli engedélye nélkül sem teljes terjedelmében, sem részleteiben semmilyen módon, illetve eszközzel nem másolható, információ-visszakereső rendszerben nem tárolható, nem mikrofilmezhető

Tartalom

Előszó ... 7
Bevezető ... 9
A könyv használatáról ... 9
Selejtező
1. Összekevert programok ... 15
Összekevert program 1. ... 16
Megoldás ... 18
A program felépítése, működése ... 18
Összekevert program 2. ... 18
Megoldás ... 20
A program felépítése, működése ... 20
Összekevert program 3. ... 21
Megoldás ... 24
A program felépítése, működése ... 25
2. Input–output feladatok ... 27
Input–output feladat 1. ... 27
Megoldás ... 28
Input–output feladat 2. ... 28
Megoldás ... 29
Input–output feladat 3. ... 30
Megoldás ... 30
3. Elrontott programok ... 31
Elrontott program 1. ... 31
Megoldás ... 36
Elrontott program 2. ... 39
Megoldás ... 44
Elrontott program 3. ... 45
Megoldás ... 50
4. Egysoros programok megírása ... 53
Megoldás ... 53
5. Tesztfeladatok 55
Tesztfeladat 1. ... 55
Megoldás 55
Tesztfeladat 2. ... 56
Megoldás 56
Tesztfeladat 3. ... 56
Megoldás 65
Tesztfeladat 4. ... 65
Megoldás 66
Tesztfeladat 5. ... 66
Megoldás ... 66
Tesztfeladat 6. ... 66
Megoldás ... 67
6. Változatok néhány szál gyufára ... 68
A program felépítése, működése ... 76
Megoldás 77

Vigasság
7. Memory 81
A program felépítése 95
A program működése 98
8. TOTÓ: mit csinál ez az egysoros program? 100
9. Hibakeresés 112
Megoldás 112
Döntő
10. SCILful és a SZKIták 115
Játékszabály 115
Megoldás 117
A SZKIták bosszúja: a programírási feladat 117
A program felépítése, működése 122
Különbségek a négy gép programjában 123
11. Tesztfeladatok 125
Megoldás 127
12. Házi feladat 128
Speciális képernyőkezelés az M08X számítógépen 128

Könyvecskénket azoknak ajánljuk, akik szeretnék próbára tenni számítástechnikai ismereteiket, BASIC tudásukat.

A Számítástechnikai Koordinációs Intézet (SzKI) és Leányvállalatai — a SCI-L és a SCITEL — 1983 után 1984-ben is megrendezték a Budapesti Nemzetközi Vásáron a számítógépes BASIC vetélkedőjüket, kettős céllal: részben hogy megismertessék személyi számítógépüket az érdeklődőkkel, a jövendő számítástechnikai szakemberekkel; részben pedig hogy segítséget adjanak a vetélkedőn részt vevők számára programozási szemléletük, feladatmegoldási készségük fejlődéséhez. A vetélkedőben elsősorban középiskolás diákok vettek részt, de a nevezettek között voltak általános iskolások, egyetemisták és profi számítógépesek is.

A vetélkedő győztesei értékes díjakat, ösztöndíjakat kaptak. Ezenkívül a különböző társadalmi és tömegszervezetek, szakmai egyesületek, fórumok képviselőiből álló társadalmi zsüri is számos különdíjat osztott szét.

A vetélkedő sikere egyértelmű volt. Mint ahogy az is kiderült, hogy az ilyenfajta játékokban a teenagerek messze felülmúlják a náluk idősebb és akár nagyobb tudású ellenfeleket. Egyfelől mert BASIC-ben valóban ők a legjobbak — hiszen a profik, néhány kivételtől eltekintve, nem ezt a nyelvet használják —, másfelől mert merészek, vállalkozó kedvűek, és nincs veszteni valójuk. No és nem elhanyagolható az a szempont sem, hogy ez a korosztály olyan kreatív képességek birtokában van, amelyek elengedhetetlenek egy értelmes vetélkedő feladatainak sikeres teljesítéséhez.

Végignézve az idei vetélkedő feladatait úgy gondoltuk, érdemes lenne azokat közzé tenni, hogy ne csak azok ismerjék, akik ott voltak a vetélkedő selejtezőjében, döntőjében. Egyrészt mert mindenki szeret játszani, s ezek a feladatok nem nélkülözik a játékosságot, másrészt mert sok, a programozás tanulása szempontjából hasznos ismeret van elrejtve ezekben a feladatokban. A feladatokat készítő, s a vetélkedő lebonyolítását, szervezését az SzKI megbízásából végző, AKCIÓ Játékszervező Gazdasági Munkaközösség tagjai és szakértői — Angyalosi László, Horváth András, Király Zoltán — arra törekedtek, hogy — az SzKI és Leányvállalatai célkitűzéseinek megfelelően — olyan feladatokat készítsenek, amelyeknek megoldása a résztvevők számítástechnikai, programozói ismereteit elmélyíti.

E könyv épp arra jó, amire a vetélkedő kevéssé adott alkalmat: a feladatok visszalapozhatók, és amire a vetélkedő során mindössze 10 perc jutott, azon most — több idő lévén — lehessen töprengeni, gondolkodni. Hiszen más a célja egy vetélkedőnek, ahol végül is ki kell választani a legjobbakat, s ehhez bizony nehéz feladatokat, s még nehezebb feltételeket (pl. a feladatok megoldására adott korlátozott időt) kell adni, és más a célja egy könyvnek, amely nem vetélkedésre szolgál, hanem okulásra.

A könyv tervének megszületésekor adódott az ötlet a szerkesztő részéről, hogy a vetélkedő minden lehetséges anyagát írassuk át olyan mikroszámítógépekre, amelyek a legelterjedtebbek hazánkban. Így a vetélkedőhöz csak M08X gépre megírt programok, feladatok átkerültek a HT1080Z, a Commodore-64 és a Sinclair Spectrum gépekre is. Ez könyvünk újabb érdekessége, hiszen ily módon az olvasónak egy sor

olyan összehasonlításra van lehetősége, amelyre eddig aligha volt. Láthatják, hogy hogyan lehet megoldani ugyanazt a feladatot, ugyanazt a problémát az egyik és a másik gépen, hogy melyik az egyszerűbb, melyik miben tud többet vagy kevesebbet. (Persze e könyv alapján mindez csak a gépek BASIC-jére vonatkozhat.)

Könyvünkhöz kívánunk valamennyi olvasónak tanulságos rejtvényfejtést és kellemes szórakozást!

A vetélkedő kiírásakor az volt a célunk, hogy minél több nevezés érkezzen, minél nagyobb tömegű versenyző vehessen részt a vetélkedő első fordulójában, a selejtezőkben. Ezért választottuk a selejtezésre az írásbeli módszert. Közel 200 nevezés érkezett, s e 200 résztvevőt négy csoportban versenyeztettük. Egy-egy selejtező kb. háromórás fárasztó munkát jelentett a versenyzők számára. Célunk az volt, hogy az írásos mód ellenére a selejtezőben is használjuk a számítógépet. Kitaláltuk tehát, hogy bizonyos feladatok esetén az írásbeli megoldáshoz a szükséges információk egy részét a gépeknél történő bemutatás adja meg. Sikerült azt is megoldanunk, hogy ha nem is sokat, de legalább 2 percet minden résztvevő eltölthesen az M08X gépeknél. Erre szolgált az a játék, amely a selejtezők első feladata volt minden csoportban.

A selejtezőkből 12-en jutottak a vetélkedő döntőjébe. A döntő maradék négy helyére a BNV-n lehetett bekerülni. Ott ugyanis az Intézet standján felállított M08X gépeken futtattunk egy olyan 15 perces programot, amely három különböző feladatot adott a versenyzőknek, s a 15 perc eltelte után mindenki egy pontszámot kapott eredményként. E pontszámok alapján a játék három napján egy-egy versenyző szintén megszerezte a jogot a döntőbe kerülésre, és a negyedik helyre a három nap legjobb pontszámot elért második helyezettje került. Itt, az ún. vigaszágban kb. 150-en vettek részt. Köztük voltak olyanok is, akik az írásbeli selejtezőn kiestek, hiszen vigaszág lévén ezt nem zárták ki a szabályok. (Azt viszont igen, hogy valaki a BNV-n kétszer ülhessen géphez!)

Feltétlenül igaz az, hogy a vetélkedő döntőjébe nehezebb volt bekerülni az igazi írásbeli selejtezőkből, mint a vigaszágról. De egyfelől ez a vigaszágnál törvényszerű, másrészt az arányok azért a vigaszágnál mégis nehezítették a résztvevők dolgát. Hiszen míg az írásbeli selejtezőből 200 versenyzőre jutott 12 döntős hely — tehát minden 17 versenyzőből egy juthatott tovább —, addig a vigaszágon ez az arány rosszabb volt, hiszen 150-ből 4-en juthattak be, azaz 37 nevezőre jutott egy hely!

Vetélkedőnkben volt némi szerepe a szerencsének is. Ez azonban nem haladta meg az egészséges arányt. Úgy gondoljuk, hogy a játékosság magával hozza a szerencse szerepét, márpedig vetélkedőt játékosság nélkül elképzelhetetlennek tartunk!

Könyvünk felépítésében megtartjuk a vetélkedő lebonyolítási rendjének megfelelő tagolást, azaz előbb közöljük az írásbeli selejtező feladatait, majd a vigaszág következik, végül pedig a döntő feladatai.

A könyv használatáról

Minden feladatot igyekeztünk közérthető formában megfogalmazni. A selejtező azon feladatai esetében, ahol a számítógép melletti bemutató a feladat megoldásához szükséges információkat adott, ezeket képernyőfotókkal pótoltuk. Reméljük sikerült!

A feladatok után közvetlenül közöljük a megoldást, tehát nem kell az olvasónak hátralapoznia a megoldások után böngészve, hanem a következő oldalon vagy lapon megtalálja azt. Viszont, hogy a feladatokat megoldani kívánók játékát se rontsuk el, egyértelmű jellel jelezzük, hol kezdődik a feladat megoldása. Ily módon elkerülhető, hogy valaki „véletlenül" elolvasson egy megoldást, mielőtt még maga megkísérelné azt megtalálni.

Amint azt könyvünk bevezetőjében is olvashatták, igyekeztünk minden arra érdemes programot áttenni az M08X számítógépről a HT1080Z-re, a Commodore-64-re és a Sinclair Spectrumra is. Ki-ki azzal a feladatmegoldással kísérletezzen tehát, amelyik gép BASIC-jét jobban ismeri!

Egyértelmű jelek választják el egymástól a négyféle gépre vonatkozó listákat, információkat mind a feladat anyaga, mind pedig megoldása esetén. Általában azonban a feladatok megfogalmazásának legelső része valamennyi gépre vonatkozik, s csak azután jön a speciális rész: lista, ábra stb. Ezeket az általános, mindegyik gépre vonatkozó részeket a négy gép neveinek betűiből alkotott, semmit sem jelentő, de jól hangzó MOHTODORUM fantázianévvel jelezzük.

A jelek a négyféle gépre és a MOHTODORUM-ra a következők:

M08X

HT1080Z

Commodore-64

Sinclair Spectrum

MOHTODORUM

Selejtező

Az írásbeli selejtezőben a feladatsor összeállításakor feltételeztük, hogy a résztvevők ismerik a BASIC programozási nyelvet. Elsősorban a HT1080Z gép BASIC-jének ismeretére építettünk, mert az M08X BASIC-je leginkább a HT1080Z-éra hasonlít.

Ami speciális, ill. eltérő van az M08X BASIC-ben, arról egy rövid összefoglalót adtunk a versenyzőknek. Ezt az összefoglalót valamennyi feladatmegoldásnál használhatták. Mivel feltételezzük, hogy olvasóink közül sem mindenki ismeri az M08X-et, ezért itt közöljük ezt a kis összefoglalót:

Néhány tudnivaló az M08X BASIC-ről

- Logikai kifejezések "igaz" értékét -1, "hamis" értékét 0 ábrázolja.

- NEXT I:NEXT J:NEXT K helyett NEXT I,J,K írható.

Utasítások:

LINE INPUT < string változó >

Teljes sor (legfeljebb 254 karakter) bevitele egy string tipusú változóba.

DEF FN < név > [(paraméter >[,<paraméter>]...)] =
< függvénymeghatározás >

Egy, a felhasználó által írt függvényt határoz meg és nevez el. A függvény hívása FN < név > (< paraméterlista >) alakban történik.

Függvények:

CHR$(I)

A függvény visszaadja az I kifejezésben megadott érték ASCII kódnak megfelelő karakterét. CHR$(17) törli a képernyőt. A nagybetűk kódja 32-vel kisebb a megfelelő kisbetűk kódjánál.

INSTR([I,]X$,Y$)

Megkeresi az Y$ string első előfordulását az X$ stringben és visszaadja azt a pozíciót, ahol azt megtalálta. Ha I nem szerepel, a keresés az X$ string első karakterétől indul. Ha az I-t megadjuk - melynek értéke 1 és 255 közötti lehet - ez lesz a keresés kezdő pozíciója. Ha Y$ nincs az X$ stringben, akkor 0-t ad vissza.

MID$(X$,I[,J])

Visszaad egy J karakter hosszúságú stringet, amely a karaktereket az X$ string I-ik elemétől tartalmazza. Az I és J értéke 1 és 255 között lehet. Ha J-t elhagyjuk, vagy ha a string I-ik karakterétől jobbra kevesebb jel van, mint a J értéke, az I-iktől kezdve valamennyi jobboldali karakter a visszaadott stringbe kerül.

SPACE$(X)

Visszaad egy szóközzel feltöltött X hosszúságú stringet. X értéke 0 és 255 között lehet.

STRING$(I,J) vagy STRING$(I,X$)

Visszaad egy I hosszúságú stringet, amelynek valamennyi karaktere a J érték ASCII kódjával, vagy az X$ string első karakterével van feltöltve.

A különböző napokra beosztott versenyzők azonos típusú feladatokat kaptak, de a típusokon belül a feladatok változtak. Teljesen egyforma nehézségű feladatokat természetesen nem sikerült kitalálnunk, de sikerült a feladatcsomagokat úgy összeállítani, hogy összességükben körülbelül azonos nehézségűek legyenek. Tehát, ha egyik típusból az egyik napra nehezebb feladat került, akkor figyeltünk arra, hogy egy másik típusból könnyebb kerüljön mellé.

Az érdekesség kedvéért azt is közöljük, hogy az egyes feladattípusokra mennyi időt adtunk. Ne feledjék tehát, hogy egy-egy feladattípusból egy napon egy szerepelt! Tehát a feladattípusok és a rájuk adott megoldási idők:

1. Összekevert program 2. Input—output feladat	e kettőre együttesen 35 perc.
3. Elrontott program 4. Egysoros program megírása	e kettőre szintén 35 perc.
5. Tesztfeladatok	10 perc.
6. Gyufás játék	10 perc felkészülés, 2 perc játékidő.

Összekevert programok 1

Ez a feladat bizonyára nem „eredeti" ötlet, bár a feladatok készítői nem találkoztak még vele. Majdnem azt hitték, hogy kitalálták egy vadonatúj dolgot, mígnem egyik ismerősük lehűtötte őket. Közölte ugyanis, hogy ő már találkozott efélével.

Ennek ellenére érdekes és jól használható feladattípusnak tartjuk. Ragyogóan alkalmas ui. arra, hogy felmérje, mennyire képes a feladat megoldója áttekinteni egy kész program struktúráját. Esetünkben e feladatnál használtuk először saját „találmányunkat": a képernyőn való bemutatást, mint a feladat bevezető lépését. A selejtezőben részt vevők odasereglettek a felállított M08X gépek képernyői elé, s megnézték a programot futás közben. Ezután kapták kézhez a program összekevert listáját, s nem volt más dolguk, mint az egyes sorok melletti kis keretekbe beírni a megfelelő sorszámot. Szokás szerint 10-esével sorszámoztak, amit persze pontveszteség nélkül meg lehetett szegni. A lényeg az volt, hogy úgy rakják sorba a megkapott programsorokat, hogy a „kész" program lefuttatva ugyanazt az eredményt hozza, mint a képernyőn látott bemutatáskor. (Természetesen az értékelés során nem futtattuk le a programot, hiszen enélkül is értékelni lehetett minden megoldást.)

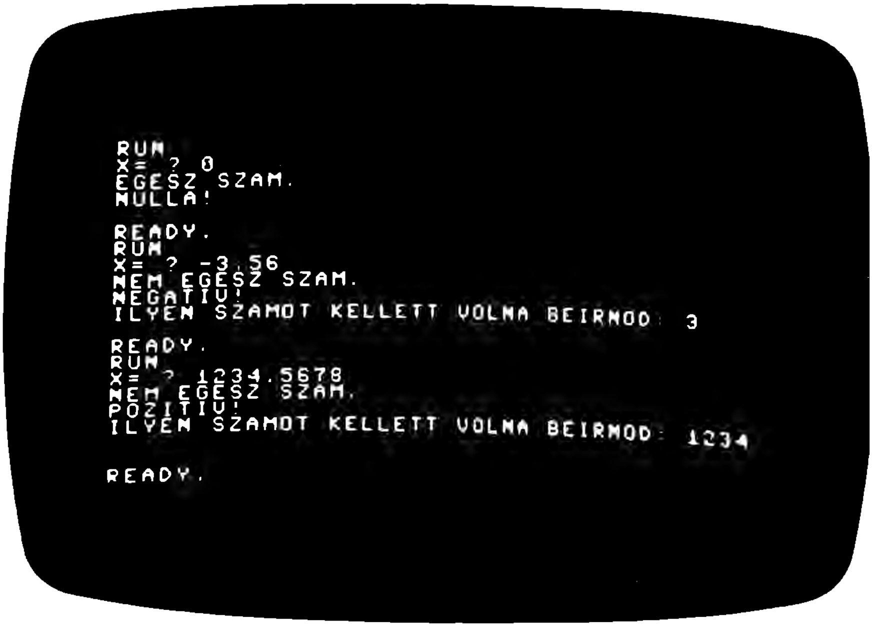
Ennek a feladatnak az értékelése során — ezt előre közöltük a versenyzőkkel — a zsüri nem ragaszkodott egyetlen helyes megoldáshoz.

Tekintettel arra, hogy könyvünkhöz nem tudunk (sajnos!) egy-egy számítógépet és egy-egy hajlékony mágneslemezt vagy kazettát mellékelni, a bemutatás helyett a képernyőképekről fotókat közlünk. Fotókat, amelyek bemutatják, hogy mit is „csinálnak" a programok. „Csinálnak!" Ugyanis ebből az összekevert feladattípusból három változat készült. Vegyük hát sorra őket.

(Az első és a második változat esetén az M08X, a HT1080Z és a Commodore-64 gépek programjai között szinte semmi eltérés nincs!)

Összekevert program 1.

A program futásáról készült képernyőkép:



1. képernyőkép
A képernyőképről a fotó kis csalással készült, ui. mivel eredetileg a képernyőre nem fért fel valamennyi futtatás egyszerre, így két képernyőképet összevágtunk egy képpé

A kitöltendő összekevert listák:

```
!__!__!__!   X=ABS(X)
!__!__!__!   PRINT"POZITIV!"
!__!__!__!   RETURN
!__!__!__!   PRINT"NEGATIV!"
!__!__!__!   PRINT"NULLA!"
!__!__!__!   RETURN
!__!__!__!   RETURN
!__!__!__!   IF X<>INT(X) THEN PRINT"NEM ";
!__!__!__!   PRINT"ILYEN SZAMOT KELLETT VOLNA BEIRNOD:";
!__!__!__!   ON (X=INT(X))*(X>=0)+1 GOTO 60,100
!__!__!__!   PRINT"EGESZ SZAM."
!__!__!__!   INPUT "X= ";X
!__!__!__!   X=INT(X)
```

```
ON 2-SGN(X) GOSUB 110,130,150
END
PRINT X
```

```
LET X=ABS X
PRINT "POZITIV!"
RETURN
PRINT "NEGATIV!"
PRINT "NULLA!"
RETURN
RETURN
IF X<>INT X THEN PRINT "NEM ";
PRINT "ILYEN SZAMOT KELLETT VOLNA BEIR- NOD:";
GO TO 60+(X=INT X)*(X>=0)*40
PRINT "EGESZ SZAM."
INPUT "X= ";X
LET X=INT X
GO SUB 130-SGN X*20
STOP
PRINT X
```

Megoldás

Egy lehetséges megoldást közlünk. Előbb felsorolásszerűen a keretekbe írandó sorszámokat, majd a teljes és működőképes listát!
A beírható sorszámok tehát
70 110-120-150-130-140-160-20-60-50-30-10-80-40-100-90

A müködőképes listák:

```
10 INPUT "X= ";X
20 IF X<>INT(X) THEN PRINT"NEM ";
30 PRINT"EGESZ SZAM."
40 ON 2-SGN(X) GOSUB 110,130,150
50 ON (X<=INT(X))*(X>=0)+1 GOTO 60,100
60 PRINT"ILYEN SZAMOT KELLETT VOLNA BEIRNOD:";
70 X=ABS(X)
80 X=INT(X)
90 PRINT X
100 END
110 PRINT"POZITIV!"
120 RETURN
130 PRINT"NULLA!"
140 RETURN
150 PRINT"NEGATIV!"
160 RETURN
```

Megoldás

```
 10 INPUT "X= ";X
 20 IF X  INT X THEN PRINT "NEM ";
 30 PRINT "EGESZ SZAM."
 40 GO SUB 130 SGN X*20
 50 GO TO 60+(X=INT X)*(X>=0)*40
 60 PRINT "ILYEN SZAMOT KELLET VOLNA BEIR- NOD:";
 70 LET X=ABS X
 80 LET X=INT X
 90 PRINT X
100 STOP
110 PRINT "POZITIV!"
120 RETURN
130 PRINT "NULLA!"
140 RETURN
150 PRINT "NEGATIV!"
160 RETURN
```

A program felépítése, működése

Amint az a futtatást bemutató képernyőképen látszik, a program lekérdez egy számot, s megállapítja róla, hogy az egész vagy sem, hogy pozitív vagy negatív. Ha a beírt szám negatív vagy nem egész, kiírja, hogy „ilyen számot kellett volna beírnod" — s kiírja a szám abszolút értékének egész részét, 0 esetén pedig kiírja, hogy egész szám és hogy nulla!

Természetesen a jó megoldások eltérhetnek az itt közölt listáktól. A 60—70—80-as sorszámú sorok cserélődhetnek, de a 80-as sor csak a 70-es után következhet. De például elképzelhető, hogy a 60-asból lesz a 80-as, a 70-esből 60-as és a 80-asból 70-es! (Természetesen a 120, a 140 és a 160-as RETURN sorok cserélődhetnek!)

Összekevert program 2.

A program futásáról készült képünk:

2. képernyőkép
Természetesen a RUN után következő első sor mindig az általunk beírt szöveg, az alatta levő pedig a program futásának az eredménye. Így pl., ha a legelső bemenő adat „semmi", akkor a program válasza egy pont. (Reméljük, a képen is látható)

A kitöltendő összekevert listák:

```
 :__:__:   FOR J=LEN(A$) TO 1 STEP -1
 :__:__:   A$=""
:__:__:__:   B$=B$+" "
:__:__:__:   IF K=0 THEN PRINT A$;" ";:GOTO 40
:  :__:__:   GOTO 40
:__:__:__:   PRINT MID$(A$,J,1);
 :__:__:   I=I+1
 :__:__:   K=1-K
:  :__:__:   NEXT J
:__:__:__:   IF I=LEN(B$) THEN PRINT ".":END
 :__:__:   PRINT" ";
 :__:__:   K=1:I=0
 :__:__:   IF A$="" THEN 50
 :__:__:   LINE INPUT B$
 :__:__:   IF MID$(B$,I,1)<>" " THEN A$=A$+MID$(B$,I,1):GOTO 50
```

Megjegyzés
A HT1080Z és a Commodore-64 gépek listái annyiban térnek el ettől, hogy a LINE INPUT helyén csak INPUT van!

```
--   FOR J=LEN A$ TO 1 STEP -1
--   LET A$=""
 -   LET B$=B$+" "
--   IF K=Ø THEN PRINT A$;" ":: GO TO 4Ø
--   GO TO 4Ø
--   PRINT A$(J);
 -   LET I=I+1
 -   LET K=1-K
     NEXT J
     IF I=LEN B$ THEN PRINT ".": STOP
     PRINT " ";
     LET K=1: LET I=Ø
     IF A$="" THEN GO TO 5Ø
     INPUT B$
     IF B$(I)<>" " THEN LET A$=A$+B$(I): GO TO 5Ø
```

Megoldás Ismét egy lehetséges megoldáshoz előbb a sorszámokat, majd a tökéletes listákat közöljük.

A beírandó sorszámok

110-40-20-100-150-120-60-90-130-50-140-30-80-10-70

A működőképes listák:

```
10 LINE INPUT B$
20 B$=B$+" "
30 K=1:I=0
40 A$=""
50 IF I=LEN(B$) THEN PRINT ".":END
60 I=I+1
70 IF MID$(B$,I,1)<>" " THEN A$=A$+MID$(B$,I,1):GOTO 50
80 IF A$="" THEN 50
90 K=1-K
100 IF K=0 THEN PRINT A$;" ";:GOTO 40
110 FOR J=LEN(A$) TO 1 STEP -1
120 PRINT MID$(A$,J,1);
130 NEXT J
140 PRINT" ";
150 GOTO 40
```

```
10 INPUT B$
20 LET B$=B$+" "
30 LET K=1: LET I=0
40 LET A$=""
50 IF I=LEN B$ THEN PRINT ".": STOP
60 LET I=I+1
70 IF B$(I)<>" " THEN LET A$=A$+B$(I): GO TO 50
80 IF A$="" THEN GO TO 50
90 LET K=1-K
100 IF K=0 THEN PRINT A$;" ";: GO TO 40
110 FOR J=LEN A$ TO 1 STEP -1
120 PRINT A$(J);
130 NEXT J
140 PRINT " ";
150 GO TO 40
```

A program felépítése, működése

Amint látható, a program egy több szóból álló szöveget úgy változtat meg, hogy az általunk megadott szavak közül minden másodikat megfordítja. A futtatás során — amint a képernyőképről kiderül — elsőnek egy üres stringet is megadtunk, azaz nem írtunk be semmit, csak megnyomtuk a RETURN gombot. Ekkor csak egy pont jelent meg a képernyőn. Ez a pont persze minden egyes futtatásnál megjelent a „mondatunk" végén!

A futtatás során használt „sziporkázó" mondatokat hosszú gyötrelmek árán sikerült összeállítanunk, s amint látható, arra törekedtünk, hogy olyan szavakat fűzzünk egybe, amelyek megfordítva is értelmes szavakat adnak. (Ettől egy esetben szándékosan eltértünk.)

A megoldásban a versenyzők az általunk közölttől annyiban térhettek el, hogy a 20-as és a 30-as sort fölcserélhették.

Először a program bekéri a B$-ba az általunk kitalált szöveget. Az M08X-en ezt nem „sima adatbevitellel" (INPUT utasítással) végzi, hanem az ún. „line bevitellel" (LINE INPUT utasítással). Ez gyakorlatilag annyi előnyt jelent a megszokott adatbevitelekhez képest, hogy a RETURN billentyű megnyomásáig minden egyes jelet

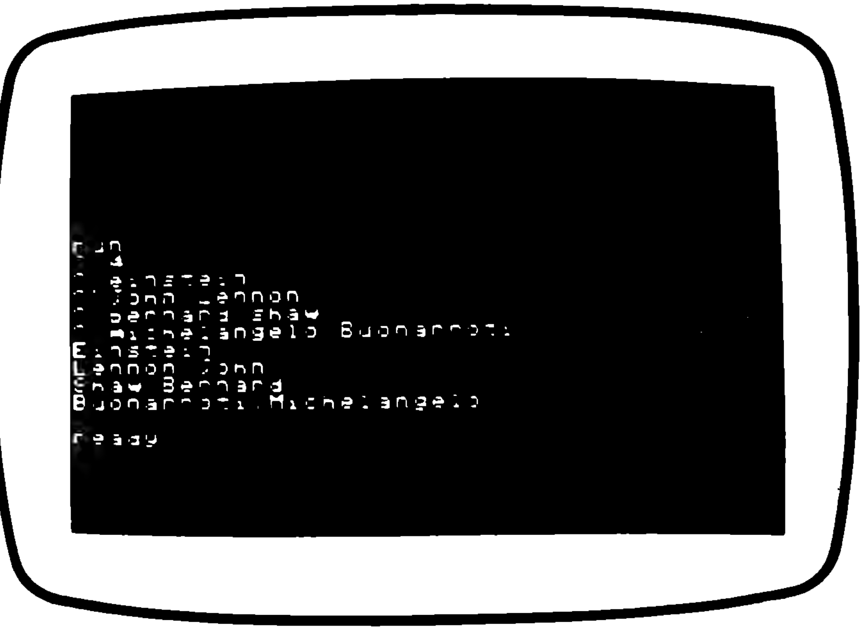
legyen az akár vessző vagy idézőjel is — a bemenő adat részeként értelmez, azaz a bekérdezendő string részének vesz.

A program a szöveg végére egy space-t illeszt. Erre azért van szükség, mert a program később, a szöveg feldolgozását szavanként végzi, s épp a szavak végén álló space-ből tudja, hogy ott egy szó vége. A programban levő K egy kapcsoló, amely azt jelzi, hogy az éppen feldolgozásra kerülő szót rendesen, avagy megfordítva kell-e kiírnia. Az I számlálja, hogy hányadik karakternél tart éppen a feldolgozás az illető (feldolgozás alatt álló) szón belül. Az A$ az éppen feldolgozás alatt levő szó tárolására szolgál. Épp ezért a kiíratás után ezt a stringet mindig kiürítjük.

A 80-as sorra azért van szükség, hogy ha netán két betűköz — space — került egymás után a szövegbe, akkor se kapjunk hibát a 120-as sorban, s a K kapcsoló értéke se változzon meg. A 110—140-es sorok ezután már „csak" elvégzik — ha szükséges — az A$-ban levő szó megfordítását, s kiírják azt a képernyőre.

Összekevert program 3.

A program futásáról készült képernyőfotó:



3 képernyőkép
Ismét a RUN utáni első sor a bemenő adat, az alatta levő adat a program futásának az eredménye

A program esetén a négyféle géphez négyféle lista tartozik. Ezek:

```
: _:  IF LEFT$(B$,1)>="a" THEN B$=CHR$(ASC(LEFT$(B$,1))-32)+MID$(B$,2)
      NEXT I
   :  B$=MID$(N$(I),K+1)
   :  C$=", "
  .:  END
```

```
|__|__|__|  PRINT B$+C$+A$
|__|__|__|  DIM N$(N)
|__|__|__|  INPUT N$(I)
|__|__|__|  INPUT N
|__|__|__|  FOR I=1 TO N
|__|__|__|  A$=LEFT$(N$(I),K-1)
|__|__|__|  NEXT I
|__|__|__|  FOR I=1 TO N
|__|__|__|  IF LEFT$(A$,1)>="a" THEN A$=CHR$(ASC(LEFT$(A$,1))-32)+MID$(A$,2)
|__|__|__|  K=INSTR(N$(I)," ")
|__|__|__|  IF K=0 THEN A$="":C$="": B$=N$(I):GOTO 130
```

```
|__|__|__|  IF LEFT$(B$,1)>="a" THEN B$=CHR$(ASC(B$)-32)+MID$(B$,2)
|__|__|__|  NEXT I
|__|__|__|  B$=MID$(N$(I),K+1)
|__|__|__|  C$=","
|__|__|__|  END
|__|__|__|  PRINT B$+C$+A$
|__|__|__|  DIM N$(N)
|__|__|__|  INPUT N$(I)
|__|__|__|  INPUT N
|__|__|__|  FOR I=1 TO N
|__|__|__|  A$=MID$(N$(I),1,K-1)
|__|__|__|  NEXT I
|__|__|__|  FOR I=1 TO N
|__|__|__|  IF LEFT$(A$,1)>="a" THEN A$=CHR$(ASC(A$)-32)+MID$(A$,2)
|__|__|__|  FOR K=1 TO LEN(N$(I))
|__|__|__|  IF MID$(N$(I),K,1)=" " THEN 100
|__|__|__|  NEXT K:A$="":C$="":B$=N$(I):GOTO 140
```

```
|__|__|__|  IF B$<"♠" THEN B$=CHR$(ASC(B$)+128)+MID$(B$,2)
|__|__|__|  NEXT I
|__|__|__|  B$=MID$(N$(I),K+1)
|__|__|__|  C$=","
```

```
END
PRINT CHR$(14);B$+C$+A$
DIM N$(N)
INPUT N$(I)
INPUT N
FOR I=1 TO N
A$=MID$(N$(I),1,K-1)
NEXT I
FOR I=1 TO N
IF A$<"^" THEN A$=CHR$(ASC(A$)+128)+MID$(A$,2)
FOR K=1 TO LEN(N$(I))
IF MID$(N$(I),K,1)=" " THEN 100
NEXT K:A$="":C$="":B$=N$(I):GOTO 140
```

```
IF B$(1)>="a" THEN  LET B$=CHR$ (CODE B$-32)+B$(2 TO )
NEXT I
LET B$=N$(I,J+1 TO K)
IF B$="" THEN  LET C$="": GO TO 190
LET C$=","
STOP
PRINT B$+C$+A$
DIM N$(N,30)
INPUT N$(I)
INPUT N
FOR I=1 TO N
LET A$=N$(I,1 TO J-1)
NEXT I
FOR I=1 TO N
IF A$(1)>="a" THEN  LET A$=CHR$ (CODE A$-32)+A$(2 TO )
FOR J=1 TO 30
IF N$(I,J)=" " THEN  GO TO 100
FOR K=30 TO 1 STEP -1
IF N$(I,K)<>" " THEN  GO TO 130
NEXT K: GO TO 200
NEXT J: LET A$="": LET C$="": LET B$=N$(I): GO TO 170
```

Megoldás Előbb ismét a keretekbe írandó sorszámok, majd a teljes működőképes listák gépenként!

130-150-120-90-160-140-20-40-10-60-100-50-30-110-70-80

140-160-130-100-170-150-20-40-10-30-110-50-60-120-70-80-90

180-200-160-170-130-210-190-20-40-10-30-140-50-60-150-70-80-100-110-120-90

A működőképes listák:

```
10 INPUT N
20 DIM N$(N)
30 FOR I=1 TO N
40 INPUT N$(I)
50 NEXT I
60 FOR I=1 TO N
70 K=INSTR(N$(I)," ")
80 IF K=0 THEN A$="":C$="": B$=N$(I):GOTO 130
90 C$=", "
100 A$=LEFT$(N$(I),K-1)
110 IF LEFT$(A$,1)>="a" THEN A$=CHR$(ASC(LEFT$(A$,1))-32)+MID$(A$,2)
120 B$=MID$(N$(I),K+1)
130 IF LEFT$(B$,1)>="a" THEN B$=CHR$(ASC(LEFT$(B$,1))-32)+MID$(B$,2)
140 PRINT B$+C$+A$
150 NEXT I
160 END
```

```
10 INPUT N
20 DIM N$(N)
30 FOR I=1 TO N
40 INPUT N$(I)
50 NEXT I
60 FOR I=1 TO N
70 FOR K=1 TO LEN(N$(I))
80 IF MID$(N$(I),K,1)=" " THEN 100
90 NEXT K:A$="":C$="":B$=N$(I):GOTO 140
100 C$=","
110 A$=MID$(N$(I),1,K-1)
120 IF LEFT$(A$,1)>="a" THEN A$=CHR$(ASC(A$)-32)+MID$(A$,2)
130 B$=MID$(N$(I),K+1)
140 IF LEFT$(B$,1)>="a" THEN B$=CHR$(ASC(B$)-32)+MID$(B$,2)
150 PRINT B$+C$+A$
160 NEXT I
170 END
```

```
10 INPUT N
20 DIM N$(N)
30 FOR I=1 TO N
40 INPUT N$(I)
50 NEXT I
60 FOR I=1 TO N
70 FOR K=1 TO LEN(N$(I))
80 IF MID$(N$(I),K,1)=" " THEN 100
90 NEXT K:A$="":C$="":B$=N$(I):GOTO 140
100 C$=","
```

```
110 A$=MID$(N$(I),1,K-1)
120 IF A$<"♠" THEN A$=CHR$(ASC(A$)+128)+MID$(A$,2)
130 B$=MID$(N$(I),K+1)
140 IF B$<"♠" THEN B$=CHR$(ASC(B$)+128)+MID$(B$,2)
150 PRINT CHR$(14);B$+C$+A$
160 NEXT I
170 END
```

```
10 INPUT N
20 DIM N$(N,30)
30 FOR I=1 TO N
40 INPUT N$(I)
50 NEXT I
60 FOR I=1 TO N
70 FOR J=1 TO 30
80 IF N$(1,J)=" " THEN GO TO 100
90 NEXT J: LET A$="": LET C$="": LET B$=N$(I): GO TO 170
100 FOR K=30 TO 1 STEP -1
110 IF N$(I,K)<>" " THEN GO TO 130
120 NEXT K: GO TO 200
130 LET C$=","
140 LET A$=N$(I,1 TO J-1)
150 IF A$(1)>="a" THEN LET A$=CHR$ (CODE A$-32)+A$(2 TO )
160 LET B$=N$(I,J+1 TO K)
170 IF B$="" THEN LET c$="": GO TO 190
180 IF B$(1)>="a" THEN LET B$=CHR$ (CODE B$-32)+B$(2 TO )
190 PRINT B$+C$+A$
200 NEXT I
210 STOP
```

A program felépítése, működése

Amint a képernyőképről láthattuk, a program amolyan könyvtári kisegítő programként működik. Az első bemenő adatként megadott szám értékének megfelelő mennyiségű nevet kér, s ezeket úgy dolgozza fel, hogy egyrészt a „véletlenül" beírt kis kezdőbetűket naggyá változtatja, majd a beírt idegen neveket a magyar névhasználat szerint megforgatja. Előre veszi a vezeték- (vagy családi) nevet, és utána illeszti a keresztnevet. A könyvtári katalogizálásban használt írásmódnak megfelelően a kettő közé egy vesszőt ír. Így lesz pl. „John Lennon"-ból „Lennon, John".

Először egy számot kér a program, s ezt *N*-ben tárolja. Majd *N* számú bemenő adatot vár, amelyeket az N$ vektorban helyez el. A bemenő adatok elraktározása után a program azokat sorraveszi. Az A$-ba helyezi el a keresztnevet, a B$-ba a családi nevet, és a C$-ba egy vesszőt. Ahhoz persze, hogy minden a helyére kerüljön, a programnak szét kell választania a név két részét. Ezt úgy teszi meg, hogy megkeresi az épp soron levő „anyagban" az első betűközt, és ebből tudja, hogy meddig tart a keresztnév. Ez a keresés az M08X gépen egy egyszerű INSTR függvénnyel, a többi gépen pedig egy ciklussal történik. (Ráadásul a Sinclair Spectrumon a dimenzionálás miatt a vektor elemei mind a megadott hosszúságúak kell hogy legyenek, így a nevek után még néhány space is áll. Így itt egy másik ciklusban meg kell keresni az utolsó nem betűköz karaktert.) Ha a program úgy találja, hogy a megadott név csak egy szóból áll, kiüríti az A$-t és a C$-t, majd azonnal arra a programrészre megy, amely a B$ első karakterét vizsgálja, és ha szükséges, ezt az első karaktert kisbetűről nagyra változtatja. Amennyiben azonban két részből áll a név, tehát a betűköz után is áll még „valami", akkor a következő történik.

Az A$-ba betölti a program a betűköz előtt álló részt, majd elvégzi a kezdőbetű vizsgálatát. A B$-ba kerül a betűköz utáni rész, s ismét a vizsgálat következik, végül pedig kiírja a B$-t, a C$-t és az A$-t.

Ezután kezdődik minden elölről, ill., ha a program az *N* darab adattal (névvel) végzett, hát jobb dolga nem lévén megáll. E feladatnak több jó megoldása is lehetséges. Egyrészt föl lehet cserélni a 30-as és a 60-as sorokat, ill. az 50-est az M08X-en, a 150-essel, a HT1080Z-n és a Commodore-64-en a 160-assal, a Sinclair Spectrumon a 190-essel. Másrészt, mint könnyen belátható, a 90—120, ill. a HT1080Z-n és a Commodore-64-en a 100—130, a Sinclair Spectrumon a 130—160-as sorok sorrendjéből csak a két középső egymásutánisága kötött.

Input—output feladatok 2

Ezt a nevet adtuk annak a feladatnak, amelyben egy rövid programot kellett végigtekinteniük a versenyzőknek, és erről a programról kellett eldönteni, hogy mit csinál. Ezután a programlista mellett megadott bemenő adatokkal számítógép nélkül el kellett végezni a programban elhelyezett műveletsort, és a kapott végeredményt kellett közölni. (Természetesen zsebszámológépek használatát engedélyeztük, sőt egy olyan versenyző is akadt, aki programozható Sharp számítógépet hozott magával. Ez ellen sem tiltakozott a zsüri, hiszen ha valaki rájött arra, hogy mit csinál a program — márpedig erre a kérdésre úgyis válaszolnia kellett —, akkor hamarabb kiszámolhatta a végeredményt, mintha bepötyögte volna a programot.)

A feladat értékelése során szomorúan tapasztaltuk, hogy a sok zsebszámológép és számítógép erősen lerontja a fiatalok számtani tudását. Helyes logikával, de rossz számolással a többség elvétette a megoldást.

Lássuk tehát a feladatok három változatát a négy géptípusra (olykor a különbség minimális)!

A kérdések

1. Mit csinál ez a program?
2. Mennyi lesz a program futásának eredménye a megadott bemenő adatokkal?

Input—output feladat 1.

```
10 DEF FN S(X,Y,S1,S2)=(S1*X+S2*Y)/(S1+S2)
20 INPUT N
30 DIM V(N)
40 FOR I=N TO 1 STEP -1:INPUT V(I):NEXT I
50 FOR I=1 TO N+RND
60 S=FNS(S,V(I),I-1,INT(RND+1))
70 NEXT I
80 PRINT S
```

```
5 GOTO 20
10 S=(S1*S+S2*Y)/(S1+S2):RETURN
20 INPUT N
30 DIM V(N)
40 FOR I=N TO 1 STEP -1:INPUT V(I):NEXT I
50 FOR I=1 TO N+RND(0)
60 S1=I-1:S2=RND(1):Y=V(I):GOSUB 10
70 NEXT I
80 PRINT S
```

```
10 DEF FN S(Y)=(S1*S+S2*Y)/(S1+S2)
20 INPUT N
30 DIM V(N)
40 FOR I=N TO 1 STEP -1:INPUT V(I):NEXT I
50 FOR I=1 TO N+RND(0)
60 S1=I-1:S2=INT(RND(0)+1):S=FN S(V(I))
70 NEXT I
80 PRINT S
```

```
10 DEF FN S(X,Y,P,Q)=(P*X+Q*Y)/(P+Q)
20 INPUT N
30 DIM V(N)
40 FOR I=N TO 1 STEP -1: INPUT V(I): NEXT I
50 LET S=0
60 FOR I=1 TO N+RND
70 LET S=FN S(S,V(I),I-1,INT (RND+1))
80 NEXT I
90 PRINT S
```

Megjegyzés: A Commodore-64 és a HT1080Z gépre írt program elsősorban azért különbözik a másik kettőtől, mert a DEF FN függvényt másképpen használja, vagy egyáltalán nem ismeri.

A bemenő adatok:

```
 11
  7
- 1
 1Ø.3
  2
- 2.3
 13
  3.62
  Ø
  8
- 1
  4.38
```

Megoldás

1. A program először az *N* értékét kéri be, majd bekér *N* darab számot, s kiszámítja ezek számtani közepét. (*Megjegyzés:* a számtani közép számítását a program nem a hagyományos módszerrel végzi. E módszemek nagy előnye — bár ezt ez esetben programunkban nem használtuk ki —, hogy nem kell tudnunk előre a bemenő adatok számát. A program ugyanis minden egyes adat beírása után tárolja az eddig megadott adatok számtani közepét.)

2. A megadott adatokkal a végeredmény: 4.

Input—output feladat 2.

```
10 INPUT A,B
20 E=SGN(A)*SGN(B):A=ABS(A):B=ABS(B)
30 HH=1:IF B*HH>A THEN 50
40 HH=10*HH:IF B*HH<A THEN 40 ELSE 60
50 HH=.1*HH:IF B*HH>A THEN 50
60 V=HH:K=1
70 FOR I=1 TO 10000
80 IF B*H>A THEN IF K=0 THEN H=H-V ELSE V=.1*V:H=H-V:K=0
90 IF B*H<A THEN IF K=1 THEN H=H+V ELSE V=.1*V:H=H+V:K=1
100 IF B*H=A OR V<.0000001*HH  THEN 120
110 NEXT I
120 PRINT E*H
```

```
10 INPUT A,B
20 E=SGN(A)*SGN(B):A=ABS(A):B=ABS(B)
30 HH=1:IF B*HH>A THEN 60
40 HH=10*HH:IF B*HH<A THEN 40
50 GOTO 60
60 HH=.1*HH:IF B*HH>A THEN 60
70 V=HH:K=1
80 FOR I=1 TO 10000
90 IF B*H<=A THEN 120
100 IF K=0 THEN H=H-V:GOTO 120
110 V=.1*V:H=H-V:K=0
120 IF B*H>=A THEN 150
130 IF K=1 THEN H=H+V:GOTO 150
140 V=.1*V:H=H+V:K=1
150 IF B*H=A OR V<.00000001*HH THEN 170
160 NEXT I
170 PRINT E*H
```

```
  10 INPUT A,B
  20 LET E=SGN A*SGN B: LET A=ABS A: LET
 B=ABS B
  30 LET H=0: LET HH=1: IF B*HH>A THEN
GO TO 60
  40 LET HH=10*HH: IF B*HH<A THEN  GO TO
 40
  50 GO TO 70
  60 LET HH=.1*HH: IF B*HH>A THEN  GO TO
 60
  70 LET V=HH: LET K=1
  80 FOR I=1 TO 10000
  90 IF B*H<=A THEN  GO TO 120
 100 IF K=0 THEN  LET H=H-V: GO TO 120
 110 LET V=.1*V: LET H=H-V: LET K=0
 120 IF B*H>=A THEN  GO TO 150
 130 IF K=1 THEN  LET H=H+V: GO TO 150
 140 LET V=.1*V: LET H=H+V: LET K=1
 150 IF B*H=A OR V<=.0000001*HH THEN  GO
 TO 170
 160 NEXT I
 170 PRINT E*H
```

Megjegyzés: A Commodore-64 és a Sinclair Spectrum BASIC-je nem ismeri az ELSE utasítást, így a programok némileg bonyolultabbak, mint a másik két gép esetében.

A bemenő adatok: 0.314
0.0001

Megoldás

1. A program beolvas két számot, majd kiszámítja és kiírja ezek hányadosát! Mindezt a program úgy végzi el, hogy nem tartalmaz egyszerű osztás műveletet, hanem előbb tárolja a hányados előjelét, majd a számok abszolút értékével számol tovább. A HH változóban megbecsüli a hányados nagyságrendjét. Erre azért van szükség, hogy pl. 12E23/34E-8 esetén is véges időn belül véget érjen a program futása, s ráadásul pontos eredményt adjon. A becsült értékből kiindulva programunk iterációval pontosítja a H hányados értékét mindaddig, amíg az eltérés elhanyagolhatóan kicsi nem lesz. Végül ezt a H-t írja ki a program, az előjellel együtt.

2. A megadott bemenő értékekkel a program futásának eredménye: 3140. (Vetélkedőnk során ezt az eredményt legtöbben „elszámolták", s 314-et írtak helyette!)

Input—output feladat 3.

```
10 INPUT "Hany szam lesz? ";N
20 DIM V(N)
30 FOR I=1 TO N
40 INPUT V(I)
50 NEXT I
60 LET TT=1E38
70 FOR I=1 TO N-1
80 FOR J=I+1 TO N
90 LET T=ABS (V(I)-V(J))
100 IF T<TT THEN  LET TT=T: LET L1=I: L
ET L2=J
110 NEXT J
120 NEXT I
130 PRINT L1,L2,,TT
```

Megjegyzés: A közölt lista a Sinclair Spectrum gép programjának listája. A többi gépen az értékadó utasításnál a LET kulcsszó szükségtelen.

A bemenő adatok: 10
0
13
—5.6
6.23
1
—4.98
7.12
—6.22
—5.1
2.25

Megoldás

1. A program előbb megkérdezi, hogy hány bemenő adat, hány szám lesz, majd ezeket bekéri és sorban betölti egy vektorba. Ezután a megadott számok közül kiválasztja az egymáshoz „legközelebb levő" kettőt, és kiíratja a sorszámukat, valamint a különbségük abszolút értékét.

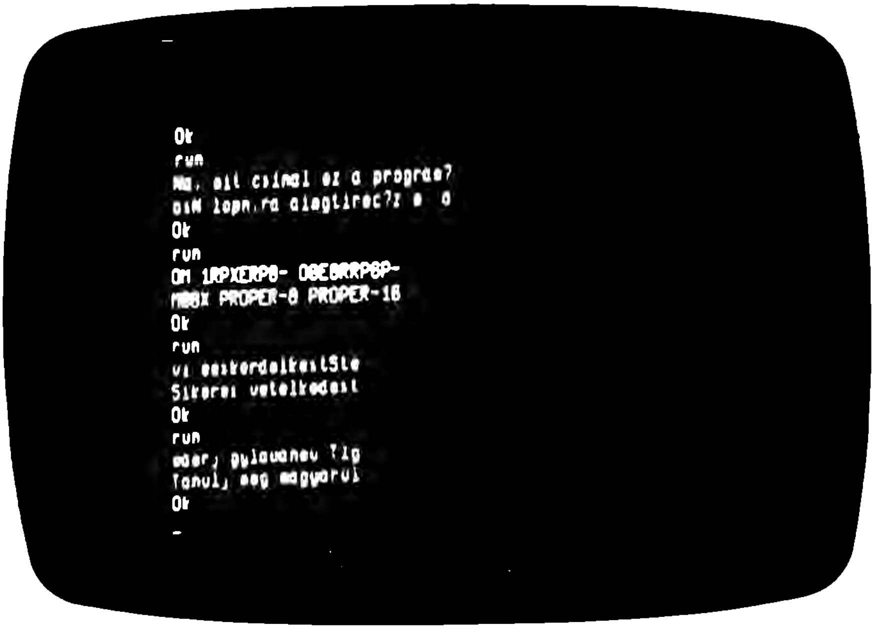
Megjegyzés: A 60-as sorban a TT kezdőértékének közel a legnagyobb megengedett számot adtuk meg, mert enélkül nem működne a program.

2. A megadott értékekkel a program futásának eredménye: 6., 9., 012.

A programhibák javítása régi, ismert — szinte már elcsépelt — feladványa a különböző programozási vetélkedőknek. Ami mégis újítást jelent e feladatok készítői részéről, az egyrészt ismét a program futásának bemutatása — az egyszerű leírás helyett —, másfelől pedig az a sokkal lényegesebb „apróság", hogy az itt elrontott programok működőképesek maradtak, csak éppen nem azt csinálták, amit kellett volna. Így az elrontott feladatok bemutatása két részben történt. Előbb lefuttattuk a versenyzők szeme láttára a „jó" programot, majd a „rosszat". Ahol a program bemenő adatokat várt, ott azok betűről betűre megegyeztek a két futtatásnál. Ily módon a versenyzők láthatták, s jegyzetelhették, hogy mi történt az egyik, s mi a másik esetben. Ezután kapták kézhez a második futtatáskor használt programot, tehát az ún. „elrontott" program listáját, s most az volt a feladatuk, hogy keressék meg, mit kell a programon változtatni, javítani ahhoz, hogy az az első futtatási eredményt produkálja ismét. Ebből a feladatból is háromféle készült.

Elrontott program 1.

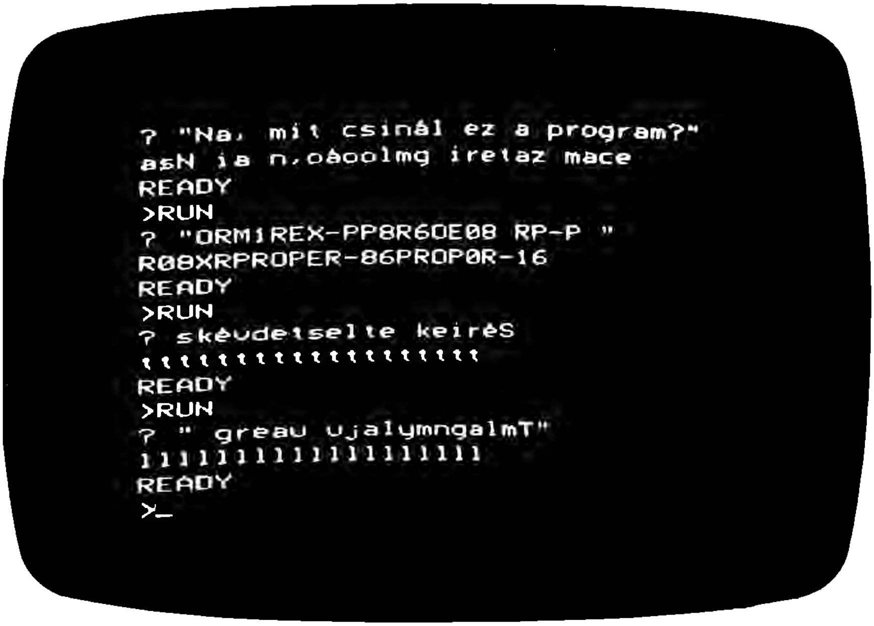
A program „jó" és „rossz" változatának futtatásáról készült képernyőképek:



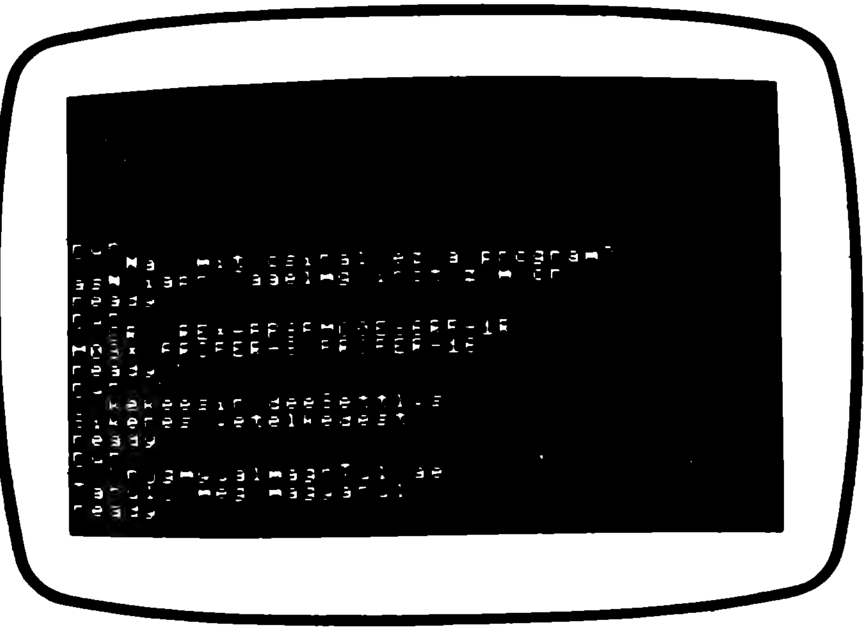
4. képernyőkép
A program jó változatának futtatása

5. képernyőkép
A program elrontott változatának futtatása

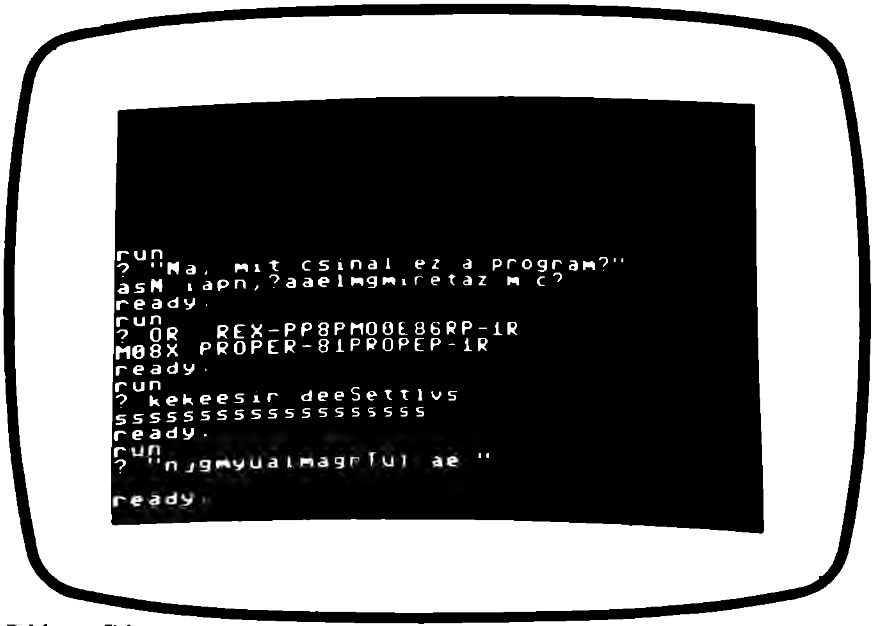
6. képernyőkép
A program jó változatának futtatása



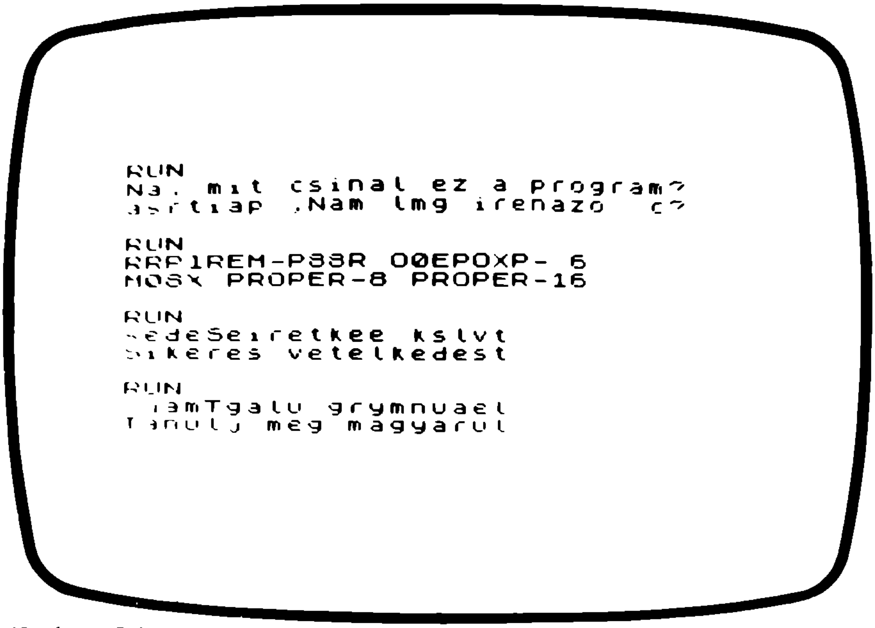
7. képernyőkép
A program elrontott változatának futtatása



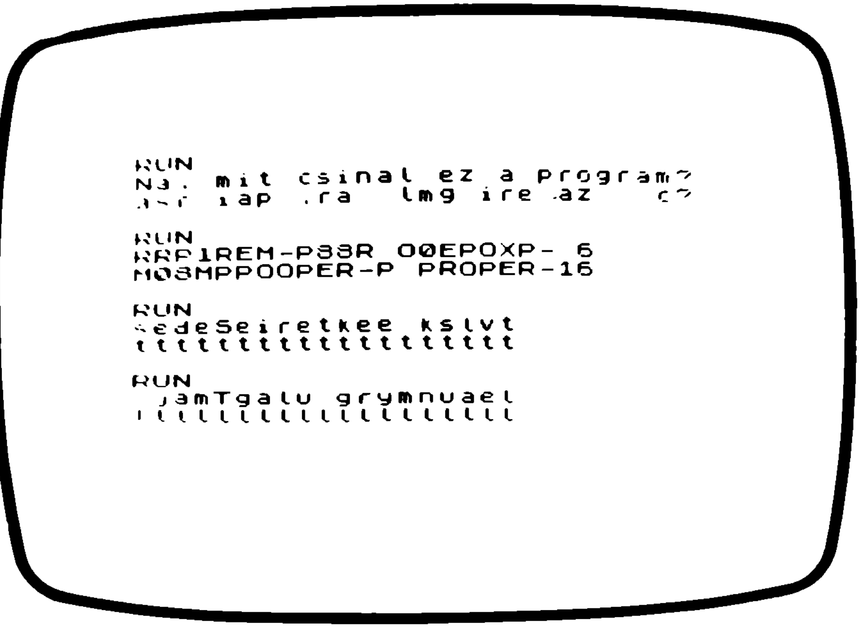
8. képernyőkép
A program jó változatának futtatása



9. képernyőkép
A program elrontott változatának futtatása



10. képernyőkép
A program jó változatának futtatása



11. képernyőkép
A program elrontott változatának futtatása. A kép elkészítése során némi csalás történt, hiszen a Spectrum csak külön „kérésre" hajlandó arra, hogy minden bemenő adat és futtatási eredmény — ráadásul még a RUN felirat is — a képernyőn maradjon. Ezt a kis csalást kérjük bocsássák meg nekünk, hiszen nem akartuk, hogy ezt az egy programfuttatást 10 képpel kelljen illusztrálnunk.
A lényeg természetesen az, hogy a RUN után beírt szöveggel mi történik.

Ezekután a programok elrontott változatainak listái:

```
10 LINE INPUT A$
20 N=LEN(A$)
30 DIM B$(N)
40 FOR I=1 TO N
50 B$(I)=MID$(A$,I,1)
60 NEXT I
70 FOR I=1 TO 5
80 P1=1+INT(RND*N)
90 P2=1+INT(RND*N)
100 C$=B$(P1)
110 B$(P1)=B$(P2)
120 NEXT I
130 P=INT(RND*3)*6+5-2*(RND<.5)
140 FOR J=1 TO N
150 SZ=SZ+P
160 IF SZ>N THEN SZ=SZ-N:GOTO 160
170 PRINT B$(SZ);
180 NEXT J
190 END
```

```
6 POKE 16530,0:R=RND(0):POKE 16530,77:CLEAR
7 POKE 16530,0:R=RND(0):POKE 16530,77:CLEAR 1000
8 POKE 16530,0:R=RND(0):POKE 16530,77
9 FOR I=1 TO 11:R=RND(0):NEXT I
10 INPUT A$
20 N=LEN(A$)
30 DIM B$(N)
```

```
40 FOR I=1 TO N
50 B$(I)=MID$(A$,I,1)
60 NEXT I
70 FOR I=1 TO 5
80 P1=1+INT(RND(0)*N)
90 P2=1+INT(RND(0)*N)
100 C$=B$(P1)
110 B$(P1)=B$(P2)
120 NEXT I
130 P=INT(RND(0)*3)*6+5-2*(RND(0)<.5)
140 FOR J=1 TO N
150 SZ=SZ+P
160 IF SZ>N THEN SZ=SZ-N:GOTO 160
170 PRINT B$(SZ);
180 NEXT J
```

```
5 R=RND(-1):FOR I=1 TO 19:R=RND(1):NEXT I
10 INPUT A$
20 N=LEN(A$)
30 DIM B$(N)
40 FOR I=1 TO N
50 B$(I)=MID$(A$,I,1)
60 NEXT I
70 FOR I=1 TO 5
80 P1=1+INT(RND(1)*N)
90 P2=1+INT(RND(1)*N)
100 C$=B$(P1)
110 B$(P1)=B$(P2)
120 NEXT I
130 P=INT(RND(1)*3)*6+5-2*(RND(1)<.5)
140 FOR J=1 TO N
150 SZ=SZ+P
160 IF SZ>N THEN SZ=SZ-N:GOTO 160
170 PRINT B$(SZ);
180 NEXT J
```

```
 10 INPUT B$
 15 RANDOMIZE 0
 20 LET N=LEN B$
 30 FOR I=1 TO 5
 40 LET P1=1+INT (RND*N)
 50 LET P2=1+INT (RND*N)
 60 LET C$=B$(P1)
 70 LET B$(P1)=B$(P2)
 80 NEXT I
 90 LET P=INT (RND*3)*6+5+2*(RND<.5)
100 LET SZ=0: FOR J=1 TO N
110 LET SZ=SZ+P
120 IF SZ>N THEN  LET SZ=SZ-N: GO TO 120
130 PRINT B$(SZ);
140 NEXT J
```

Megoldás

A program felépítése, működése: Természetesen ahhoz, hogy a hibát megtaláljuk, végig kell követnünk a program működését.

A program bekér egy szöveget, s ezt az A$-ban, a szöveg hosszát pedig az *N* vál-

tozóban tárolja. Ezután a szöveget — a Sinclair Spectrum program kivételével — a B$ vektorba töltjük. Elkezdődik a betűk keverése.

Először a program ötször egymás után véletlenszerűen kijelöl két-két karaktert (P1, P2), és ezeket megcseréli. Majd előállít egy 5—19 közötti prímszámot: a *P*-t. Megvizsgálja, hogy ez osztója-e *N*-nek, ha igen, akkor új számot keres. Ezután következik a keverésnek az a módja, amelynek során a program veszi a szöveg *P*-edik karakterét, ezt kiírja, majd ettől a karaktertől számlál *P*-t, s a talált karaktert ismét kiírja. *P*-értékű számlálás esetén a betűsor végéről annak elejére térünk vissza, s ott folytatjuk a számlálást. Ezt a műveletet *N*-szer végzi el programunk, azaz anynyiszor, ahány karakterből áll a szöveg. Tekintettel arra, hogy a *P* prímszám és nem osztója *N*-nek, így minden karakterre csak egyszer léphet programunk. Tapasztalatok szerint ez a módszer elég jó keverést jelent.

A program tehát semmi mást nem csinál, mint a beadott szöveg betűit összevissza keveri. Hogy lehet az, hogy mégis értelmes szövegek keletkeznek a program futtatása után, legalábbis a program „jó" változata esetén?

Ez már csak a feladat kitalálóinak játékos kedvét mutatja. Az M08X gépnek ui. (hiszen eredetileg erre készült a program) van egy olyan tulajdonsága, hogy az RND függvény minden RUN után ugyanazokat a véletlenszámokat generálja. Csak vissza kellett fejteni, hogy a keverés során melyik betű hová kerül, s így némi bogarászással sikerült olyan értelmetlen betűhalmazokat összeállítani, amelyek a futtatás során a kívánt értelmes szöveget adták. A következőkben látható programokat megvizsgálva azután az is kiderül, hogy ugyanez már lényegesen bonyolultabb volt a másik három gépre való átírásnál. De mert a feladatok készítői szerették volna, hogy csacskaságaik a többi gépen is „megéljenek", megkeresték hát a módszert, hogy ez megoldható legyen. Így hát a Sinclair Spectrum program 15-ös, a Commodore-64 program 5-ös és a HT1080Z program 6—9-es sorai csak e csalafintaság létrehozását segítik.

Ezenkívül még egy lényeges eltérés van a Sinclair Spectrum programban. (Ez később még többször fog eltérést jelenteni a programok között.) A Sinclair Spectrum BASIC-ben ui. — eltérően más BASIC-ektől — a logikai IGAZ értéke +1, míg másutt −1. Ez a fontos különbség több helyen okoz előjeleltérést.

Ezenkívül többször kihasználtuk a Sinclair Spectrum programokban a gép BASIC-jének azt a jó tulajdonságát, hogy stringen belül indexeléssel lehet az egyes karakterekre hivatkozni, ezért nincs szükség a Sinclair Spectrum gép esetén a stringvektorba töltésre.

Mi a hiba? Az előbbi működési leírás segítségével már könnyen kideríthető, hogy mi a hiba a programokban.

Egyrészt kivettük a 115-ös (a Sinclair Spectrumon a 75-ös) sort, ami a karaktercseréléseknél a harmadik lépést végzi: azaz a P2-edik karaktert átrakja a P1-edikbe. Ez tehát nem történik meg, viszont a P1-edik karakter előtte átkerül a P2-edikbe!

A másik, sokkal nehezebben felfedezhető hiba a 135-ös (a Sinclair Spectrumon 95-ös) sor kihagyása volt. Ebben a sorban végeztük el azt az említett vizsgálatot, hogy a *P* prímszám osztója-e *N*-nek. Ezt a hibát nehéz volt megtalálni, bár ezt kívánta segíteni az általunk tudatosan választott példaszöveg, amely éppen 19 betűs. Tudva ugyanis, hogy az M08X-en épp 19 lesz a gép által generált prímszám, nyilvánvaló volt, hogy ha a szöveg épp ilyen hosszúságú, úgy a program futtatása után a szöveg egyetlen karakterét kapjuk vissza 19-szer egymás után. (Ezt az előbb említett 5-ös, ill. 6—9-es sorok a többi gépen is garantálják!)

A kijavított, most már hibátlan listák:

```
10 LINE INPUT A$
20 N=LEN(A$)
30 DIM B$(N)
40 FOR I=1 TO N
50 B$(I)=MID$(A$,I,1)
60 NEXT I
70 FOR I=1 TO 5
80 P1=1+INT(RND*N)
90 P2=1+INT(RND*N)
100 C$=B$(P1)
110 B$(P1)=B$(P2)
115 B$(P2)=C$
120 NEXT I
130 P=INT(RND*3)*6+5-2*(RND(.5)
135 IF N=P*INT(N/P) THEN 130
140 FOR J=1 TO N
150 SZ=SZ+P
160 IF SZ>N THEN SZ=SZ-N:GOTO 160
170 PRINT B$(SZ);
180 NEXT J
190 END
```

```
6 POKE 16530,0:R=RND(0):POKE 16530,77:CLEAR
7 POKE 16530,0:R=RND(0):POKE 16530,77:CLEAR 1000
8 POKE 16530,0:R=RND(0):POKE 16530,77
9 FOR I=1 TO 11:R=RND(0):NEXT I
10 INPUT A$
84 N=LEN(A$)
30 DIM B$(N)
40 FOR I=1 TO N
50 B$(I)=MID$(A$,I,1)
60 NEXT I
70 FOR I=1 TO 5
80 P1=1+INT(RND(0)*N)
90 P2=1+INT(RND(0)*N)
100 C$=B$(P1)
110 B$(P1)=B$(P2)
115 B$(P2)=C$
120 NEXT I
130 P=INT(RND(0)*3)*6+5-2*(RND(0)(.5)
135 IF N=P*INT(N/P) THEN 130
140 FOR J=1 TO N
150 SZ=SZ+P
160 IF SZ>N THEN SZ=SZ-N:GOTO 160
170 PRINT B$(SZ);
180 NEXT J
```

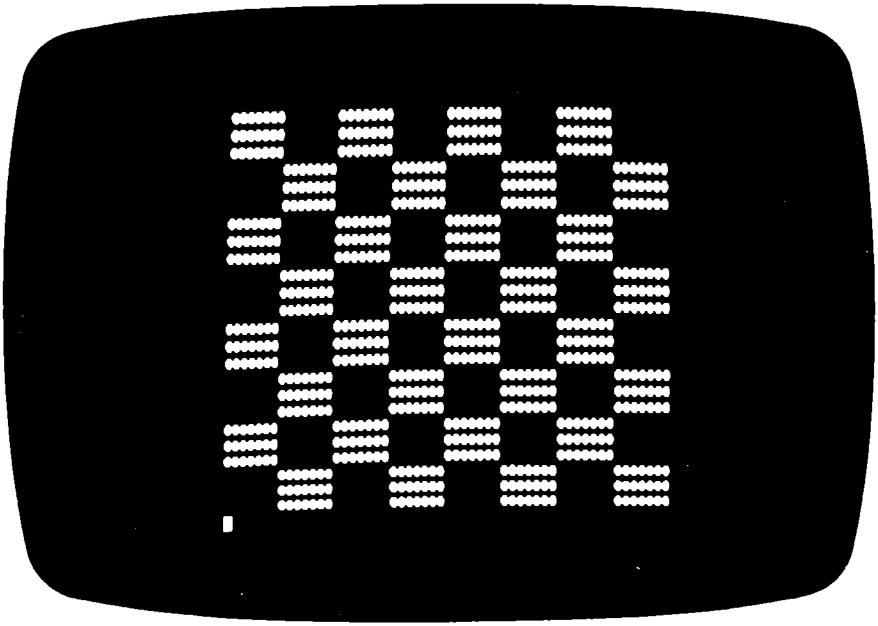
```
5 R=RND(-1):FOR I=1 TO 19:R=RND(1):NEXT I
10 INPUT A$
20 N=LEN(A$)
30 DIM B$(N)
40 FOR I=1 TO N
50 B$(I)=MID$(A$,I,1)
60 NEXT I
70 FOR I=1 TO 5
80 P1=1+INT(RND(1)*N)
90 P2=1+INT(RND(1)*N)
100 C$=B$(P1)
110 B$(P1)=B$(P2)
115 B$(P2)=C$
120 NEXT I
130 P=INT(RND(1)*3)*6+5-2*(RND(1)<.5)
135 IF N=P*INT(N/P) THEN 130
```

```
140 FOR J=1 TO N
150 SZ=SZ+P
160 IF SZ>N THEN SZ=SZ-N:GOTO 160
170 PRINT B$(SZ);
180 NEXT J
```

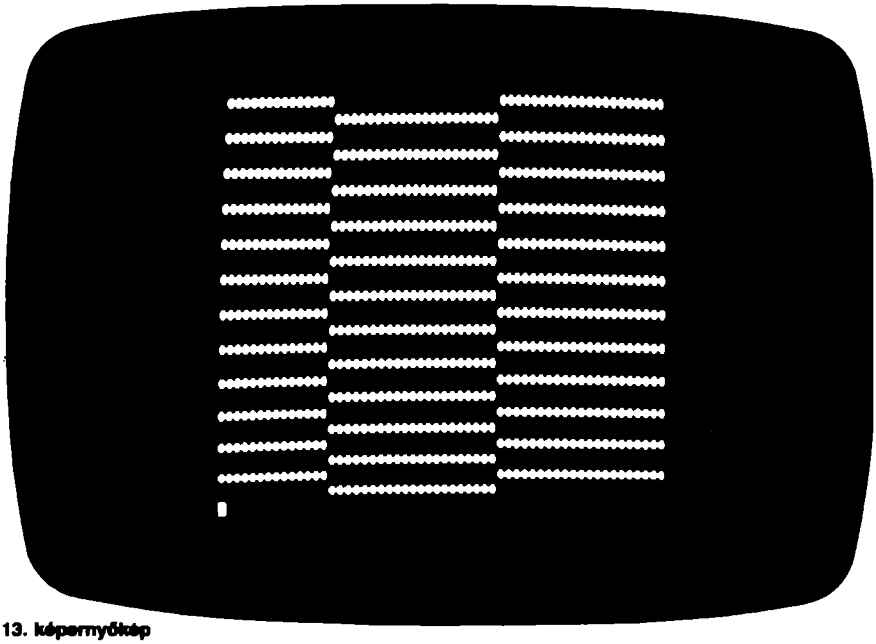
```
10 INPUT B$
15 RANDOMIZE 0
20 LET N=LEN B$
30 FOR I=1 TO 5
40 LET P1=1+INT (RND*N)
50 LET P2=1+INT (RND*N)
60 LET C$=B$(P1)
70 LET B$(P1)=B$(P2)
75 LET B$(P2)=C$
80 NEXT I
90 LET P=INT (RND*3)*6+5+2*(RND<.5)
95 IF N=P*INT (N/P) THEN GO TO 90
100 LET SZ=0: FOR J=1 TO N
110 LET SZ=SZ+P
120 IF SZ>N THEN LET SZ=SZ-N: GO TO 120
130 PRINT B$(SZ);
140 NEXT J
```

Elrontott program 2.

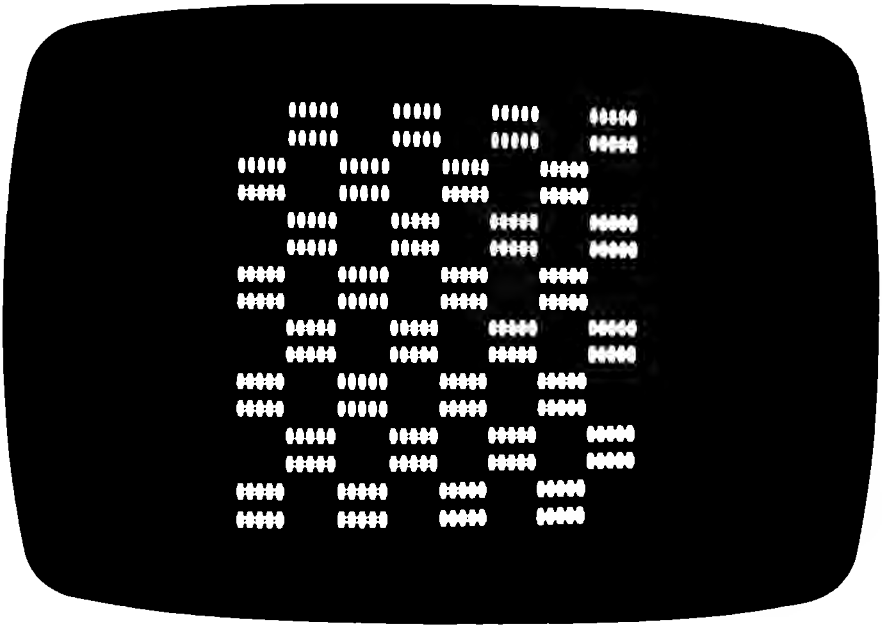
A program „jó" és „rossz" változatának futtatásáról készült képernyőképek:



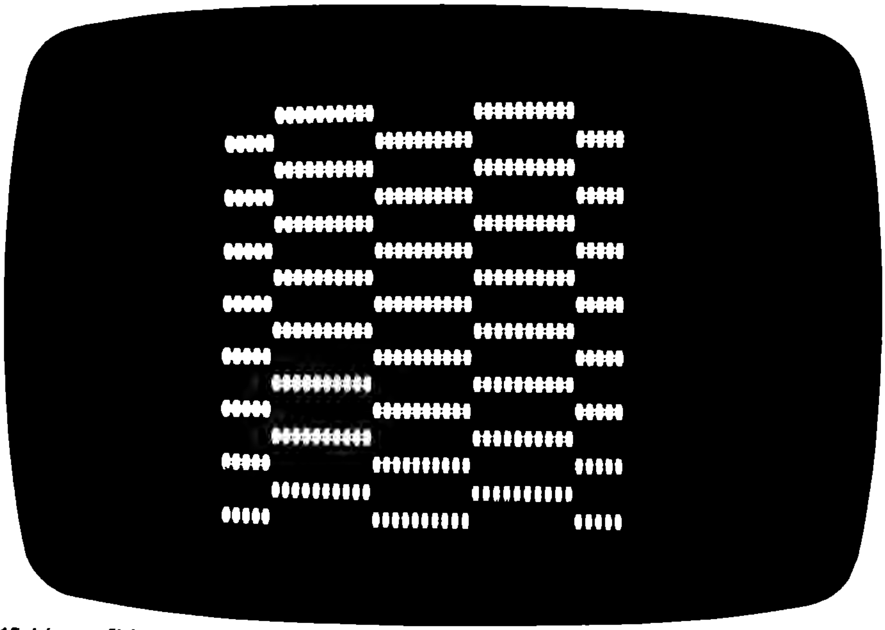
12. képernyőkép
A program jó változatának futtatása



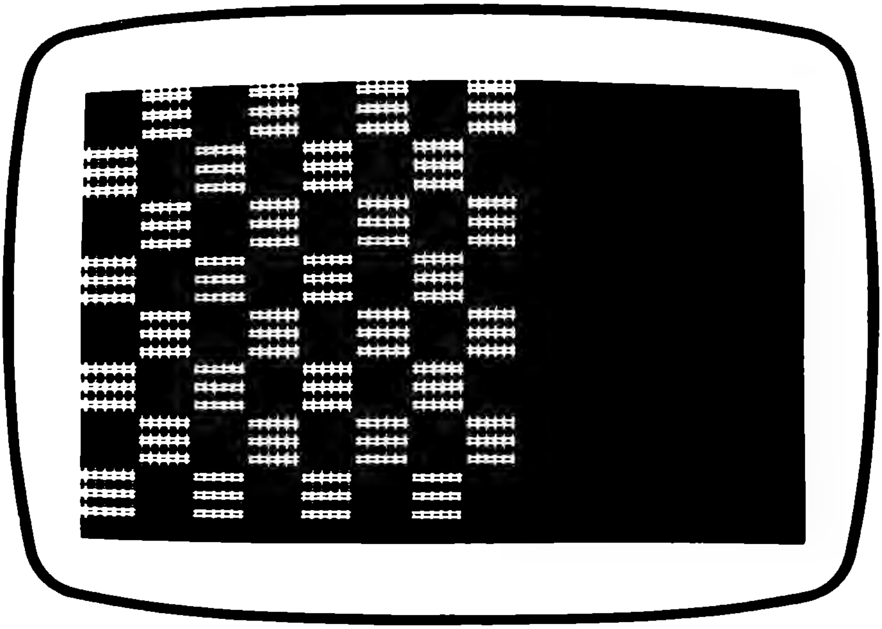
13. képernyőkép
A program elrontott változatának futtatása



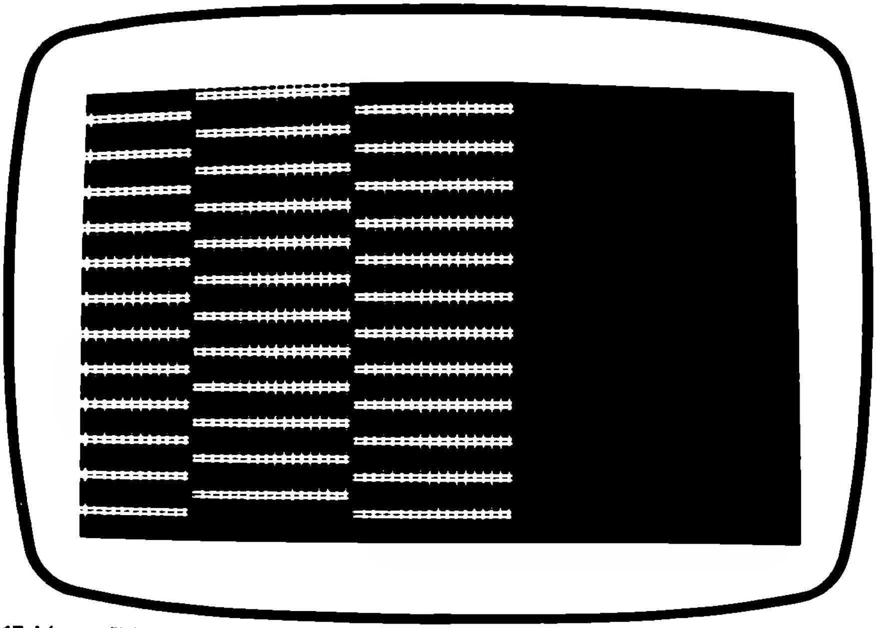
14. képernyőkép
A program jó változatának futtatása



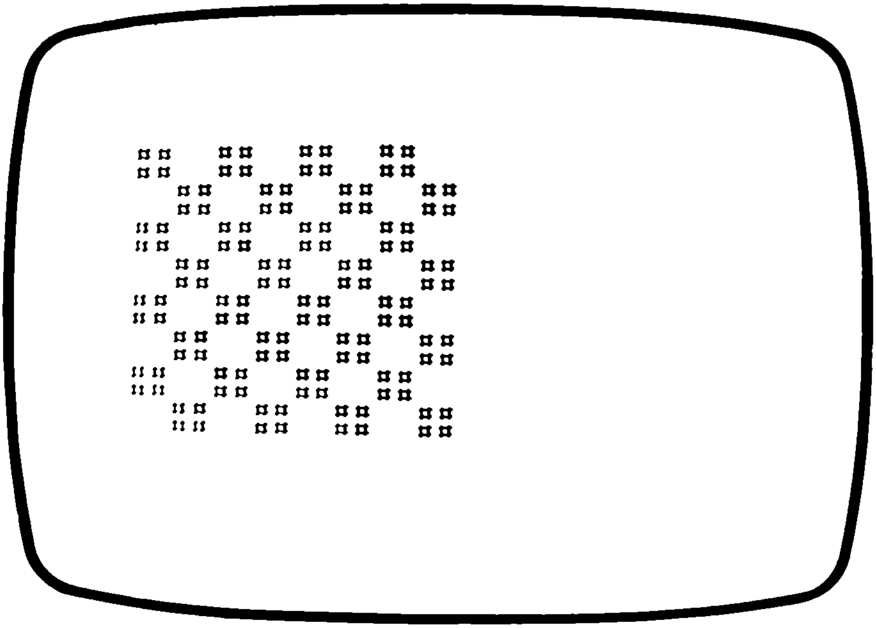
15. képernyőkép
A program elrontott változatának futtatása



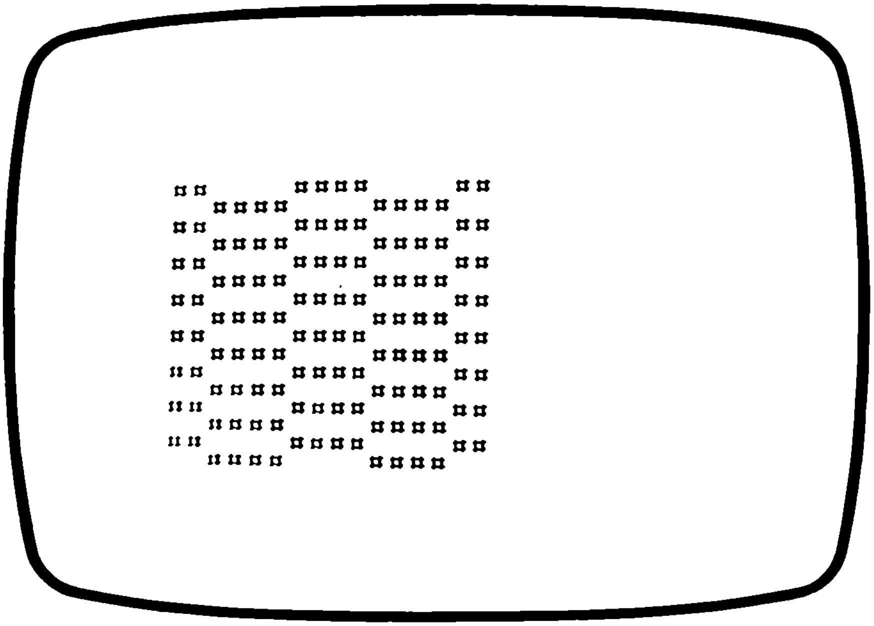
16. képernyőkép
A program jó változatának futtatása



17. képernyőkép
A program elrontott változatának futtatása



18. képernyőkép
A program jó változatának futtatása



19. képernyőkép
A program elrontott változatának futtatása

A program elrontott változatának listái:

```
10 A$(1)=STRING$(7,"#"):A$(2)=SPACE$(7)
20 PRINT CHR$(17)
30 FOR J=1 TO 24
40 FOR I=1 TO 8
50 PRINT A$(((J AND 1) = (INT(I/3) AND 1))+2);
60 NEXT I
70 PRINT
80 NEXT J
90 IF INKEY$="" THEN 90
100 END
```

```
10 A$(1)="#####":A$(2)="     "
20 CLS
30 FOR J=1 TO 16
40 PRINT
45 FOR I=1 TO 8
50 PRINT A$(((J AND 1)=(INT(I/2) AND 1))+2);
70 NEXT I
80 NEXT J
90 Q$=INKEY$:IF Q$="" THEN 90
```

```
10 A$(1)="###":A$(2)="   "
20 PRINT "]"
30 FOR J=1 TO 24
40 FOR I=1 TO 8
50 PRINT A$(((J AND 1)=(INT(I/3) AND 1))+2);
60 NEXT I
70 PRINT
80 NEXT J
90 GET G$:IF G$="" THEN 90
```

```
  10 DIM A$(2,2): LET A$(1)="##": LET A$
(2)="  "
  20 DEF FN P(X)=X-2*INT (X/2): CLS
  30 FOR J=1 TO 16
  40 FOR I=1 TO 8
  50 PRINT A$((FN P(J)=FN P(INT (I/2)))+
1);
  60 NEXT I
  70 PRINT
  80 NEXT J
  90 IF INKEY$="" THEN GO TO 90
```

Megoldás

A program felépítése, működése: Amint a fotók mutatják, a program mindössze egy sakktáblát rajzol a képernyőre, ill. a „rossz" változatban ez a tábla némiképp eltorzul. A fotók azt a kis különbséget is mutatják, ami az egyes gépek programjai között szükségképpen adódik.

A program lelke minden esetben az 50-es sor. A sorban található zárójeles kifejezés 1-et vagy 2-t ad eredményül attól függően, hogy éppen fehér vagy fekete négyzetet kell-e rajzolni, vagyis tehát ettől az értéktől függ, hogy mit rajzol a program. Ebben a sorban felhasználtuk, hogy a Sinclair Spectrum BASIC kivételével a többi az AND műveletet két tetszőleges szám között is el tudja végezni, mégpedig úgy, hogy bitenként végzi az AND műveletet. Mivel a Sinclair Spectrum ezt nem tudja, így itt a paritásvizsgálatra egy külön függvényt kellett definiálnunk (20-as sor).

Mi a hiba? Egyrészt az, hogy felcseréltük a két ciklusváltozót, másrészt az I értékét 0 helyett 1-től futtattuk. Ez utóbbit kevesen vették észre, pedig lényeges, mert ha ezt nem tesszük rendbe, akkor a sakktábla felső sorának első karaktersora a képernyő aljára kerül.

A kijavított, most már hibátlan listák:

```
10 A$(1)=STRING$(7,"#"):A$(2)=SPACE$(7)
20 PRINT CHR$(17)
30 FOR I=0 TO 23
40 FOR J=1 TO 8
50 PRINT A$(((J AND 1) = (INT(I/3) AND 1))+2);
60 NEXT J
70 PRINT
80 NEXT I
90 IF INKEY$="" THEN 90
100 END
```

```
10 A$(1)="#####":A$(2)="     "
20 CLS
30 FOR I=0 TO 15
40 PRINT
50 FOR J=1 TO 8
60 PRINT A$(((J AND 1)=(INT(I/2+ AND 1))+2);
70 NEXT J
80 NEXT I
90 Q$=INKEY$:IF Q$="" THEN 90
```

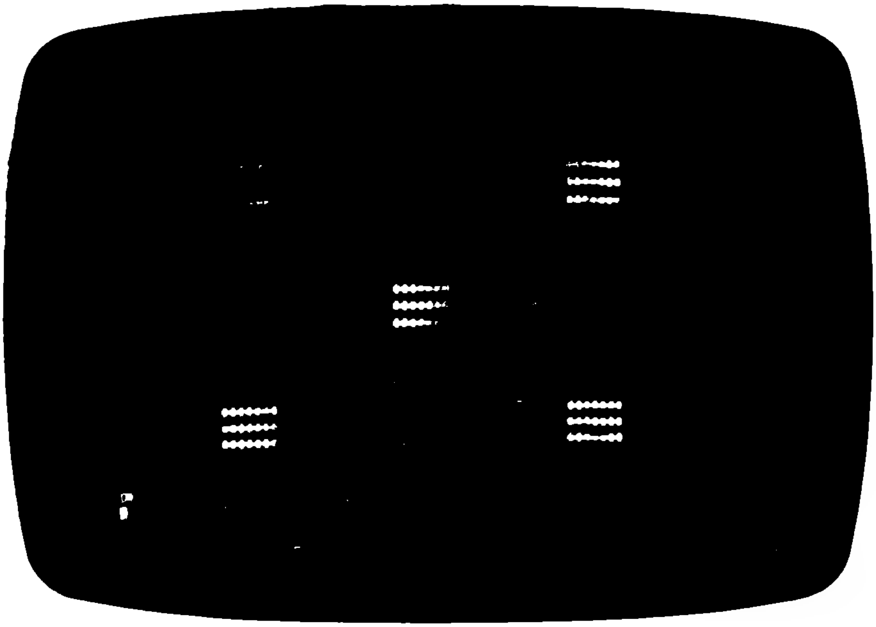
```
10 A$(1)="###":A$(2)="   "
20 PRINT "♥"
30 FOR I=0 TO 23
40 FOR J=1 TO 8
50 PRINT A$(((J AND 1)=(INT(I/3) AND 1))+2);
60 NEXT J
70 PRINT
80 NEXT I
90 GET G$:IF G$="" THEN 90
```

```
10 DIM A$(2,2): LET A$(1)="##": LET A$
(2)="  "
20 DEF FN P(X)=X-2*INT (X/2): CLS
30 FOR I=0 TO 15
40 FOR J=1 TO 8
50 PRINT A$((FN P(J)=FN P(INT (I/2)))+
1):
60 NEXT J
70 PRINT
80 NEXT I
90 IF INKEY$="" THEN GO TO 90
```

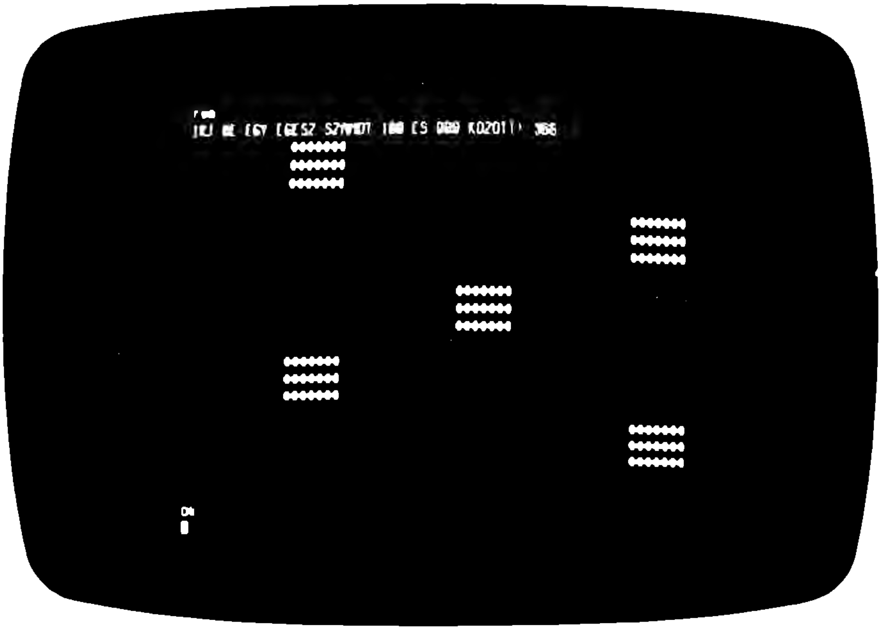
Elrontott program 3.

A program futtatása az M08X gépen a RUN után egy szám megadásával kezdődik, míg a többi gépen csak egyszerűen RUN parancsot kell adni, a többit a gép maga végzi el. Ez a különbség ismét az RND különbségeiből adódik. Az M08X-nél ahhoz, hogy különböző véletlenszámok keletkezzenek, szükség van valamilyen — az RND-t „megkeverő" — számra, a többi gépnél nincs. Ez egyben azt is jelenti, hogy míg az M08X-nél megoldható volt, hogy a „jó", ill. a „rossz" program is ugyanazokat az ábrákat rajzolja föl jól, ill. rosszul, ugyanez a másik három gépnél nem biztosítható. (A fotók készítésekor azonban kivártuk az éppen megfelelő ábrát.)

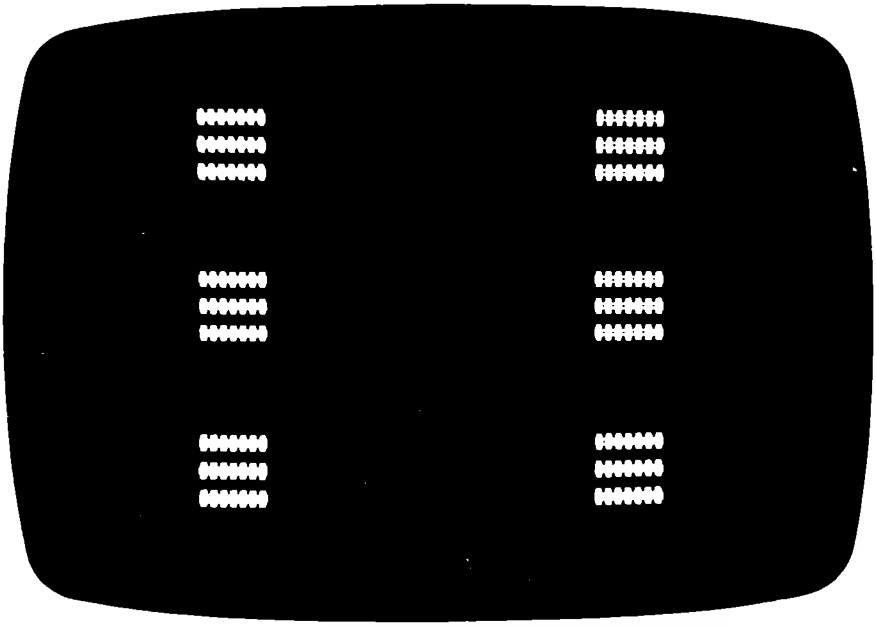
A program „jó" és „rossz" változatának futtatásáról készült képernyőképek:



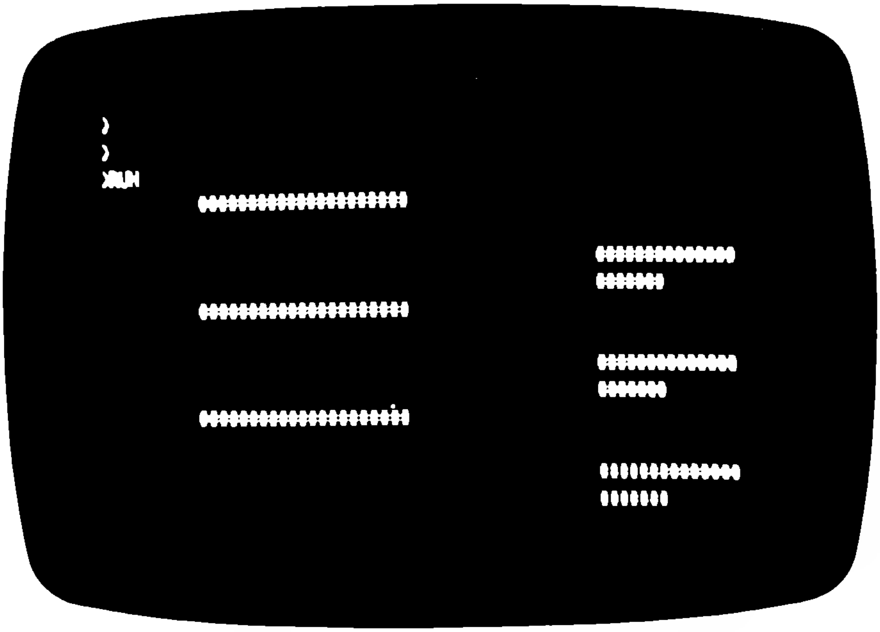
20. képernyőkép
A program jó változatának futtatása. A képen már nem látszik (mert nem fért a képernyőre), hogy a program indítása után bemenő adatot kért, amely 366 volt



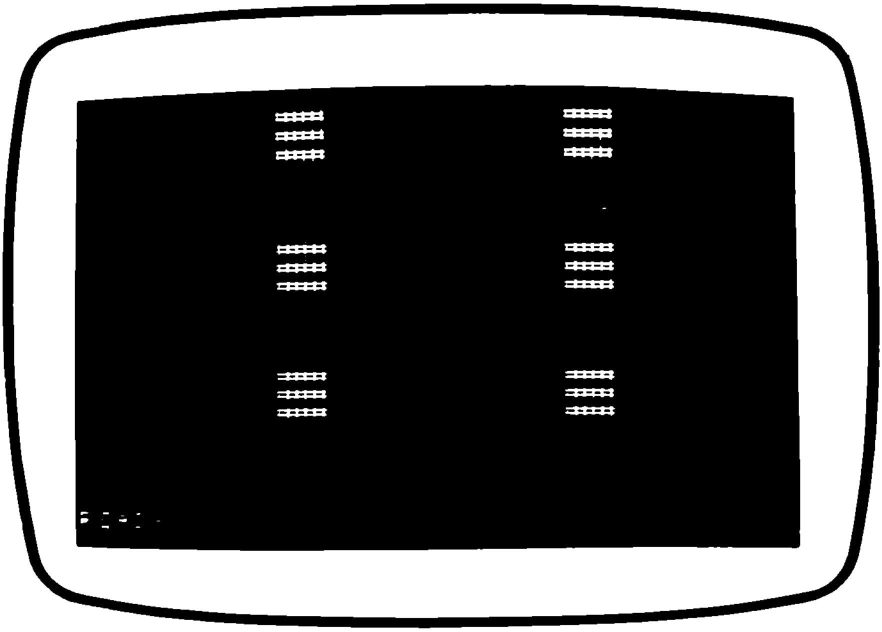
21. képernyőkép
A program elrontott változatának futtatása. A képen már nem látszik (mert nem fért a képernyőre), hogy a program indítása után bemenő adatot kért, amely 366 volt



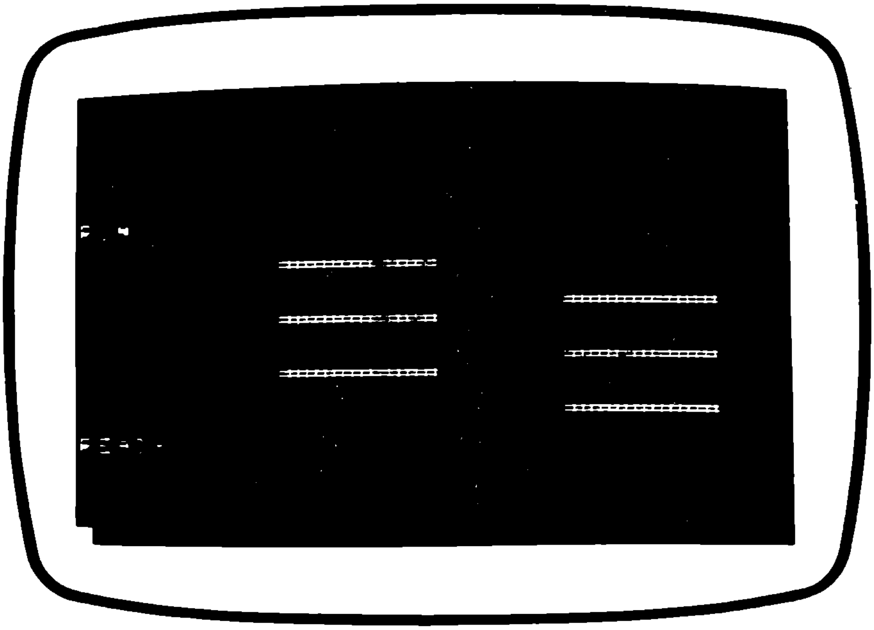
22. képernyőkép
A program jó változatának futtatása



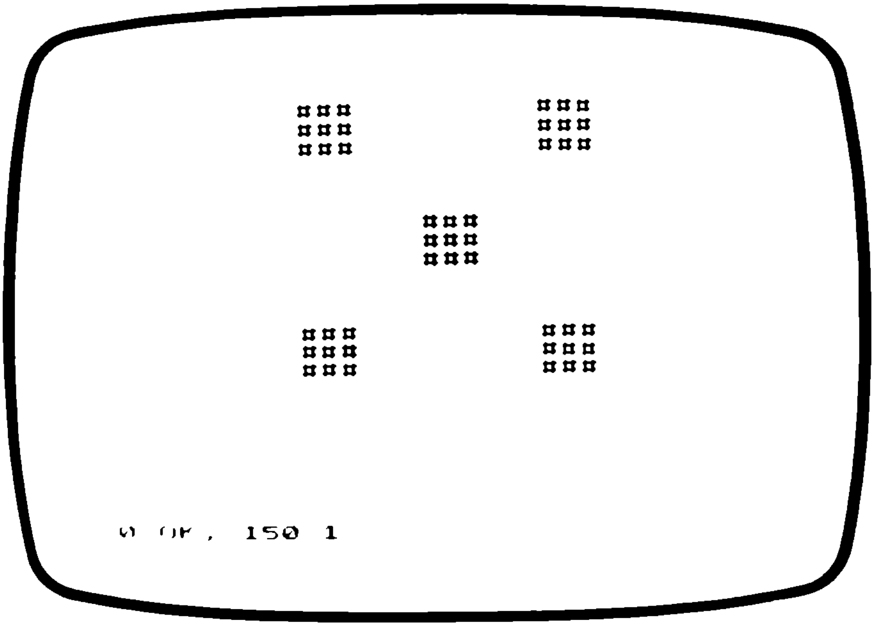
23. képernyőkép
A program elrontott változatának futtatása



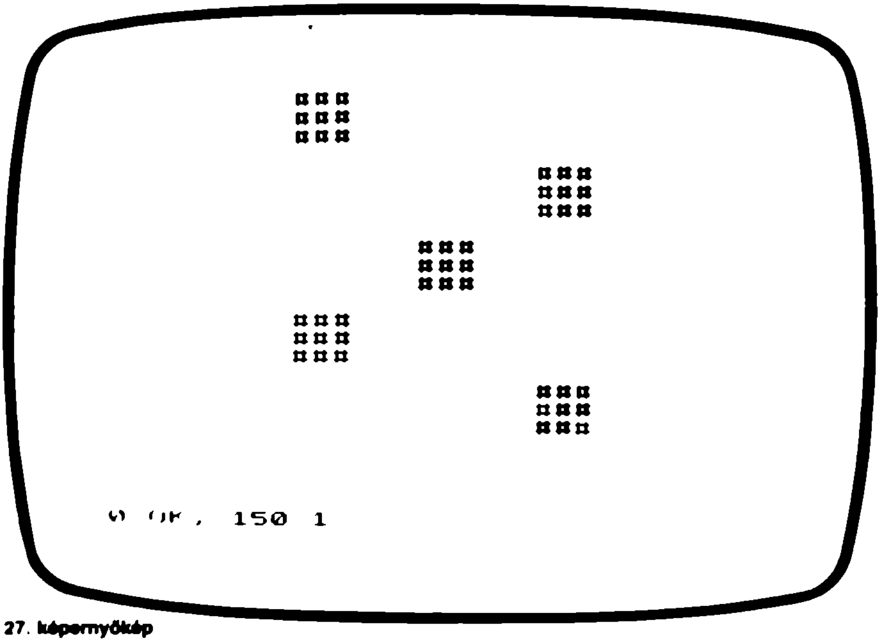
24. képernyőkép
A program jó változatának futtatása



25. képernyőkép
A program elrontott változatának futtatása



26. képernyőkép
A program jó változatának futtatása



27. képernyőkép
A program elrontott változatának futtatása

A program elrontott változatának listái:

```
10 DATA 0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,1,0,1,0,1,0,0
20 DATA 1,0,1,0,0,0,1,0,1,1,0,1,0,1,0,1,0,1,1,0,1,1,0,1,1,0,1
30 DIM T(6,3,3)
40 FOR I=1 TO 6:FOR J= 1 TO 3: FOR K=1 TO 3
50 READ T(I,J,K)
60 NEXT K,J,I
70 INPUT "IRJ BE EGY EGESZ SZAMOT 100 ES 999 KOZOTT! ",R
80 RANDOMIZE R
90 D=1+INT(6*RND)
100 FOR I=1 TO 3
110 FOR J=1 TO 3
120 FOR K=1 TO 3
130 PRINT TAB(J*21-7);LEFT$("#######",7*T(D,I,J));
140 NEXT K
150 PRINT
160 NEXT J
170 NEXT I
180 PRINT:PRINT:END
```

```
10 DATA 0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,1,0,1,0,1,0,0,1,0,1,0,0,0,1,0,1,
1,0,1,0,1,0,1,0,1,1,0,1,1,0,1,1,0,1
20 DIM T(6,3,3)
30 FOR I=1 TO 6:FOR J=1 TO 3:FOR K=1 TO 3
40 READ T(I,J,K)
50 NEXT K,J,I
60 D=1+INT(RND(0)*6)
70 FOR I=1 TO 3
80 FOR J=1 TO 3
90 FOR K=1 TO 3
100 PRINT TAB(J*20-10);LEFT$("#######",7*T(D,I,J));
```

```
110 NEXT K
120 PRINT
130 NEXT J
140 NEXT I
150 IF INKEY$="" THEN 150
```

```
10 DATA0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,1,0,1,0,1,0,0
20 DATA 1,0,1,0,0,0,1,0,1,1,0,1,0,1,0,1,0,1,1,0,1,1,0,1,1,0,1
25 PRINT
30 DIM T(6,3,3)
40 FOR I=1 TO 6:FOR J=1 TO 3:FOR K=1 TO 3
50 READ T(I,J,K)
60 NEXT K,J,I
70 D=1+INT(RND(0)*6)
80 FOR I=1 TO 3
90 FOR J=1 TO 3
100 FOR K=1 TO 3
110 PRINT TAB(J*8+3);LEFT$("###",3*T(D,I,J));
120 NEXT K
130 PRINT
140 NEXT J
150 NEXT I
```

```
  10 DATA 0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,
1,0,0,0,0,1,0,1,0,1,0,0
  20 DATA 1,0,1,0,0,0,1,0,1,1,0,1,0,1,0,
1,0,1,1,0,1,1,0,1,1,0,1
  30 DIM T(6,3,3)
  40 FOR I=1 TO 6: FOR J=1 TO 3: FOR K=1
 TO 3
  50 READ T(I,J,K)
  60 NEXT K: NEXT J: NEXT I
  70 LET D=1+INT (6*RND)
  80 FOR I=1 TO 3
  90 FOR J=1 TO 3
 100 FOR K=1 TO 3
 110 PRINT TAB (J*6+3);"###"( TO 3*T(D,I
,J));
 120 NEXT K
 130 PRINT
 140 NEXT J
 150 NEXT I
```

Megoldás

A program felépítése, működése: A program működéséről sokat nem érdemes mondani. A lényeg, hogy a program nem tesz mást, mint „kockát dob", azaz 1—6 közötti egész számot generál véletlenszerűen, s a dobásnak megfelelő ábrát fölrajzolja a képernyőre. A program elején a DATA utasításokban tároljuk a kocka egyes oldalain elhelyezkedő „pöttyöket". Egy-egy lapot kilenc négyzetre osztottunk, s a DATA utasításban sorfolytonosan helyeztük el, hogy hol van „pötty" (1-es), ill. hol nincs (0-s). A véletlenszám-generátor segítségével elvégezzük a dobást, majd kirajzoltatjuk az ábrát.

Mi a hiba? Egyrészt a K változó helyett az M08X-en a 130-as, a HT1080Z-n a 100-as, a Commodore-64-en és a Sinclair Spectrumon a 110-es sorban mindenütt a J változót írtuk be. Másrészt kihagytuk a FOR I... utáni (M08X, HT1080Z), ill. a Commodore-64-en és a Sinclair Spectrumon a NEXT I előtti PRINT utasításokat. Mint az ábrákból látható, a különböző gépek nem egyformán reagáltak az azonos hibákra. Ennek érdekes oka van. A különböző gépek ui. nem viselkednek egyformán, ha egy PRINT

TAB utasításban az éppen aktuális pozíció előtti helyre akarjuk küldeni a kurzort. Tehát, ha pl. a 17. pozíción állva a programból olyan utasítás érkezik, hogy a következő karaktert a 12. pozícióra helyezze el, akkor a Commodore-64 és a HT1080Z ott marad a 17. pozíción és oda írja, a Sinclair Spectrum és az M08X viszont sort vált, és a következő sor 12. pozíciójára helyezi el a kívánt karaktert.

Megoldás

A kijavított, most már hibátlan programlisták:

```
10 DATA 0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,1,0,1,0,1,0,0
20 DATA 1,0,1,0,0,0,1,0,1,1,0,1,0,1,0,1,0,1,1,0,1,1,0,1,1,0,1
30 DIM T(6,3,3)
40 FOR I=1 TO 6:FOR J= 1 TO 3: FOR K=1 TO 3
50 READ T(I,J,K)
60 NEXT K,J,I
70 INPUT "IRJ BE EGY EGESZ SZAMOT 100 ES 999 KOZOTT! ",R
80 RANDOMIZE R
90 D=1+INT(6*RND)
100 FOR I=1 TO 3
105 PRINT:PRINT:PRINT:PRINT
110 FOR J=1 TO 3
120 FOR K=1 TO 3
130 PRINT TAB(K*21-7);LEFT$("#######",7*T(D,I,K));
140 NEXT K
150 PRINT
160 NEXT J
170 NEXT I
180 PRINT:PRINT:END
```

```
10 DATA 0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,1,0,1,0,1,0,0,1,0,1,0,0,0,1,0,1,
1,0,1,0,1,0,1,0,1,1,0,1,1,0,1,1,0,1
20 DIM T(6,3,3)
30 FOR I=1 TO 6:FOR J=1 TO 3:FOR K=1 TO 3
40 READ T(I,J,K)
50 NEXT K,J,I
60 D=1+INT(RND(0)*6)
70 FOR I=1 TO 3
75 PRINT:PRINT:PRINT
80 FOR J=1 TO3
90 FOR K=1 TO 3
100 PRINT TAB(K*20-10);LEFT$("#######",7*T(D,I,K));
110 NEXT K
120 PRINT
130 NEXT J
140 NEXT I
150 IF INKEY$="" THEN 150
```

```
10 DATA0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,1,0,1,0,1,0,0
20 DATA 1,0,1,0,0,0,1,0,1,1,0,1,0,1,0,1,0,1,1,0,1,1,0,1,1,0,1
25 PRINT
30 DIM T(6,3,3)
40 FOR I=1 TO 6:FOR J=1 TO 3:FOR K=1 TO 3
50 READ T(I,J,K)
60 NEXT K,J,I
70 D=1+INT(RND(0)*6)
80 FOR I=1 TO 3
90 FOR J=1 TO 3
100 FOR K=1 TO 3
110 PRINT TAB(K*8+3);LEFT$("###",3*T(D,I,K));
120 NEXT K
130 PRINT
140 NEXT J
145 PRINT:PRINT:PRINT:PRINT
150 NEXT I
```

```
  10 DATA 0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,
1,0,0,0,0,1,0,1,0,1,0,0
  20 DATA 1,0,1,0,0,0,1,0,1,1,0,1,0,1,0,
1,0,1,1,0,1,1,0,1,1,0,1
  30 DIM T(6,3,3)
  40 FOR I=1 TO 6: FOR J=1 TO 3: FOR K=1
 TO 3
  50 READ T(I,J,K)
  60 NEXT K: NEXT J: NEXT I
  70 LET D=1+INT (6*RND)
  80 FOR I=1 TO 3
  90 FOR J=1 TO 3
 100 FOR K=1 TO 3
 110 PRINT TAB (K*6+3);"###"( TO 3*T(D,I
,K));
 120 NEXT K
 130 NEXT J
 140 PRINT : PRINT : PRINT : PRINT
 150 NEXT I
```

A csalafinta programozóknak szántuk ezt a feladattípust. Megadtuk egy program első sorát, a másodikat pedig elkezdtük. Ezután leírtuk, hogy mit kell tudnia a célul tűzött programnak, de kikötöttük, hogy a 20-as sorban a PRINT utasításon kívül más utasítás nem szerepelhet. Ilyen helyzet persze ritkán adódik az igazi programozási munka közben, de mindenesetre érdekes feladat végiggondolni, s megtalálni a megoldásokat.

A feladat

„Fejezd be a következő programocskák 20-as sorát úgy, hogy az ott levő PRINT utasításon kívül más utasítást nem használhatsz! (Új sort nem kezdhetsz és kettőspontot sem használhatsz!) Azt, hogy az egyes programoknak mit kell tudnia, alább olvashatod:"

a) 10 INPUT X
20 PRINT
Megállapítja egy beírt számról, hogy a szám egész vagy sem. Egész szám esetén az EGESZ, nem egész esetén a NEM EGESZ szöveg jelenik meg a képernyőn.

b) 10 INPUT X
20 PRINT
Ha a bemenő adat 1—6 közötti egész szám, akkor a szám négyzetét írja ki a képernyőre, más bemenő adatok esetén pedig 0-t.

c) 10 INPUT X
20 PRINT
A beírt szám abszolút értékének egy tizedesre kerekített megfelelőjét kapjuk a program végeredményeként akár pozitív, akár negatív számot adunk be. Ráadásul mindezt az abszolútérték-függvény használata nélkül végzi a program!

d) 10 INPUT A, B, C
20 PRINT
Ha az általunk megadott A, B, C értékek alkalmasak egy háromszög létrehozására, úgy a program kiszámítja a háromszög kerületét, ha a háromszög „nem jön össze", úgy 0 a program végeredménye.

Megoldás Az M08X, a HT1080Z és a Commodore-64 gépeken ugyanaz a megoldás működik mind a négy esetben; a Sinclair Spectrum eltér ettől.

a)

```
10 INPUT X
20 PRINT LEFT$("NEM ",-4*(X<>INT(X)))+"EGESZ'"
```

```
10 INPUT X
20 PRINT "NEM "( TO 4*(X<>INT X))+"EGESZ"
```

---

b)

```
10 INPUT X
20 PRINT -(X=INT(X))*(X>=1)*(X<=6)*X*X
```

```
10 INPUT X
20 PRINT (X>=1)*(X<=6)*(X=INT X)*X*X
```

---

c)

```
10 INPUT X
20 PRINT INT(10*X*SGN(X)+.5)/10
```

```
10 INPUT X
20 PRINT INT (10*X*SGN X+.5)/10
```

---

d)

```
10 INPUT A,B,C
20 PRINT (A+B+C)0)*(A+B>C)*(A+C>B)*(B+C>A)*(A+B+C)
```

```
  10 INPUT A,B,C
  20 PRINT (A*B*C>0)*(A+B>C)*(A+C>B)*(B+
C>A)*(A+B+C)
```

Tesztfeladatok 5

A selejtező azon feladatát neveztük tesztnek, amely egy — hat rövid feladatot tartalmazó — lapocska volt, és amelyre viszonylag rövid (10 perces) időt adtunk. A tesztfeladatok nagyobbik része olyan, hogy vagy nem kellett, vagy nem volt érdemes más géptípusokra átími, hiszen egy szintaktikus hibákat tartalmazó sor átírása egyik gépről a másikra voltaképpen azt jelenti, hogy másik sort kell ími. Így hát a tesztből elsősorban az ötleteket érdemes ellesni, hiszen ezeket a feladattípusokat ki-ki maga is folytathatja, megtöltheti tartalommal.

A tesztből négy sorozatot készítettünk. Volt olyan kérdés, amelyik mindegyikben megegyezett, de volt olyan is, amelyikből négyféle készült. Úgy gondoltuk, hogy különösen a feladványok fejtésében élvezetet találók kedvéért érdemes valamennyi elkészült feladatot közölnünk. A tesztfeladatok (amelyeket a következőkben ismertetünk) közül két feladatot: az 1-est és a 3-ast írtuk át a többi gépre. A megoldások megkönnyítésére ajánljuk föllapozni azt a rövid kis magyarázatot, amely az M08X gép BASIC-jének néhány speciális tulajdonságát ismerteti, s amelyet a vetélkedőn részt vevő versenyzők is kézhez kaptak s használhattak (l. a 13. oldalt).

Tesztfeladat 1.

A kérdés a következő:

Mire használható a következő egyszerű program?

```
10 A$=INKEY$:IF A$="" THEN 10 ELSE PRINT ASC(A$):GOTO 10
```

```
10 GET A$:IF A$="" THEN 10
20 PRINT ASC(A$):GOTO 10
```

```
10 LET A$=INKEY$: IF A$="" THEN  GO TO 10
20 PRINT CODE A$: GO TO 10
```

Megoldás

A program egy billentyű lenyomására vár. Ha ez megtörtént, kiírja a billentyű ASCII kódját, és egy újabb billentyű lenyomására vár.

Tesztfeladat 2.

Ez a feladat is három változatban készült el. Mindegyik tesztlapon a következő szöveg volt:

„Az alábbi program 20-as sorát egyszerűsítsd úgy, hogy összesen egy PRINT utasítás szerepeljen benne! Ezenkívül semmilyen más utasítás nem szerepelhet, értékadás sem!"

a)

```
10 INPUT A
20 B=1000*A:X=(B+A)/13/11:PRINT X/7
```

b)

```
10 INPUT A
20 B=A*A*A:C=B*A*A:X=C*C*B*A*A:PRINT X*X*C*B*A*A
```

c)

```
10 INPUT A
20 B=(A+2*3)/2*3-9:PRINT 4*B/3
```

Megoldás

```
a) 20 PRINT A

b) 20 PRINT A^40

c) 20 PRINT 2*A
```

Tesztfeladat 3.

A tesztfeladat 1. mellett ez a másik, amelyiknek elkészítettük a gépekre adaptált programjait. Az eredeti tesztlapon közöltünk három rövid programot, s mellette három „képernyőábrát". A versenyzők feladata az volt, hogy az egyes ábrákat párosítsák össze a megfelelő programokkal. Tekintettel azonban arra, hogy a négy különböző tesztlapon nem 4×3 ábrát használtunk, hanem 5 ábrát és a hozzájuk tartozó programokat variáltuk, úgy gondoltuk: jobb, ha itt, könyvecskénkben — kicsit nehezítve is ezzel a feladatot — közöljük az öt programot és az öt „ábrát", s ezeket kell összepárosítgatni egymással. Mindezt pedig gépenként külön-külön, hisz az egyes gépek adottságai miatt még az „ábrák" sem sikerülhettek teljesen egyformára.

Az öt program római számokkal jelölve:

```
      I.

100 INPUT T
110 FOR I=1 TO T
120 PRINT TAB(T); STRING$(2*T,"*")
130 NEXT I:END
```

II.

```
200 INPUT TT
210 FOR J=1 TO TT
220 PRINT TAB(TT-J);STRING$(2*TT,"*")
230 NEXT J:END
```

III.

```
300 INPUT L
310 FOR LL=1 TO L
320 PRINT TAB(LL),STRING$(2*L,"*")
330 NEXT LL:END
```

IV.

```
200 INPUT TT
210 FOR J=1 TO TT
220 PRINT TAB(J);STRING$(2*TT,"*")
230 NEXT J:END
```

V.

```
300 INPUT L
310 FOR LL=1 TO L
320 PRINT TAB(3*LL),STRING$(2*L,"*")
330 NEXT LL:END
```

Az öt ábra pedig:

A) változat:

```
? 13
                **************************
                **************************
                **************************
                **************************
                **************************
                **************************
                **************************
                **************************
                **************************
                **************************
                **************************
                **************************
                **************************
```

B) változat:

```
? 20
                  ****************************************
                 ****************************************
                ****************************************
               ****************************************
              ****************************************
             ****************************************
            ****************************************
           ****************************************
          ****************************************
         ****************************************
        ****************************************
       ****************************************
      ****************************************
     ****************************************
    ****************************************
   ****************************************
  ****************************************
 ****************************************
****************************************
****************************************
```

C) változat:

```
? 17
              **********************************
              **********************************
              **********************************
              **********************************
              **********************************
              **********************************
              **********************************
              **********************************
              **********************************
              **********************************
              **********************************
              **********************************
              **********************************
              **********************************
                            **********************************
                            **********************************
                            **********************************
```

D) változat:

```
? 18
************************************
 ************************************
  ************************************
   ************************************
    ************************************
     ************************************
      ************************************
       ************************************
        ************************************
         ************************************
          ************************************
           ************************************
            ************************************
             ************************************
              ************************************
               ************************************
                ************************************
                 ************************************
```

```
F) változat:

? 13
                        **************************
                        **************************
                        **************************
                        **************************
                                      **************************
                                      **************************
                                      **************************
                                      **************************
                                      **************************
                                                    **************************
                                                    **************************
                                                    **************************
                                                    **************************
```

A HT1080Z és a Commodore Spectrum listái és „ábrái” alig különböznek egymástól. A programokat külön-külön, az ábrákat azonban együtt közöljük.

A programok római számokkal jelölve:

I.

```
90 CLEAR 200
100 INPUT T
105 FOR I=1 TO 2*T:S$=S$+"*":NEXT I
110 FOR I=1 TO T
120 PRINT TAB(T);S$
130 NEXT I
```

II.

```
190 CLEAR 200
200 INPUT TT
205 FOR I=1 TO TT*2:S$=S$+"*":NEXT I
210 FOR J=1 TO TT
220 PRINT TAB(TT-J);S$
230 NEXT J
```

III.

```
290 CLEAR 200
300 INPUT L
305 FOR I=1 TO L*2:S$=S$+"*":NEXT I
310 FOR LL=1 TO L
320 PRINT TAB(LL*2);S$
330 NEXT LL
```

IV.

```
190 CLEAR 200
200 INPUT TT
205 FOR I=1 TO TT*2:S$=S$+"*":NEXT I
210 FOR J=1 TO TT
220 PRINT TAB(J);S$
230 NEXT J
```

V.

```
290 CLEAR 200
300 INPUT L
305 FOR I=1 TO L:S$=S$+"*":NEXT I
310 FOR LL=1 TO L
320 PRINT TAB(LL*4);S$
330 NEXT LL
```

I.

```
100 INPUT T
105 FOR I=1 TO 2*T:S$=S$+"*":NEXT I
110 FOR I=1 TO T
120 PRINT TAB(I);S$
130 NEXT I
```

II.

```
200 INPUT TT
205 FOR I=1 TO TT*2:S$=S$+"*":NEXT I
210 FOR J=1 TO TT
220 PRINT TAB(TT-J);S$
230 NEXT J
```

III.

```
300 INPUT L
305 FOR I=1 TO L*2:S$=S$+"*":NEXT I
310 FOR LL=1 TO L
320 PRINT TAB(LL*2);S$
330 NEXT LL
```

IV.

```
200 INPUT TT
205 FOR I=1 TO TT*2:S$=S$+"*":NEXT I
210 FOR J=1 TO TT
220 PRINT TAB(J);S$
230 NEXT J
```

```
300 INPUT L
305 FOR I=1 TO L:S$=S$+"*":NEXT I
310 FOR LL=1 TO L
320 PRINT TAB(LL*4);S$
330 NEXT LL
```

Az ábrák mindkét géphez:

```
A)

? 12

 ************************
  ************************
   ************************
    ************************
     ************************
      ************************
       ************************
        ************************
         ************************
          ************************
           ************************
            ************************
```

```
B)

? 7

*******
*******
          *******
          *******
          *******
                    *******
                    *******
```

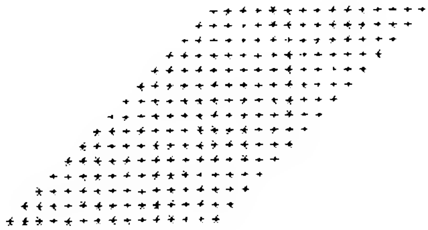
```
C)

? 8

****************
****************
****************
****************
          ****************
          ****************
          ****************
          ****************
```

```
D)

? 12

************************
************************
************************
************************
************************
************************
************************
************************
************************
************************
************************
************************
```

```
E)

? 12

           ************************
          ************************
         ************************
        ************************
       ************************
      ************************
     ************************
    ************************
   ************************
  ************************
 ************************
************************
```

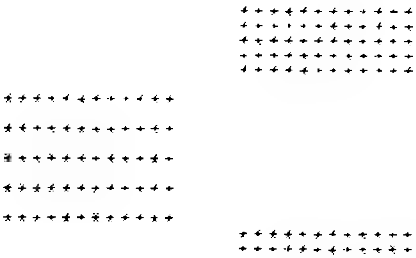
Az ábrák:

A)



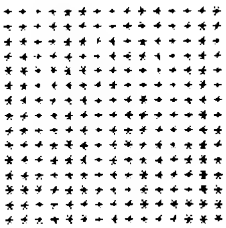
15

B)



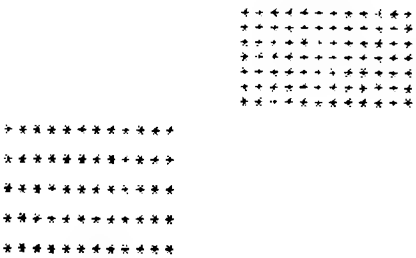
12

C)



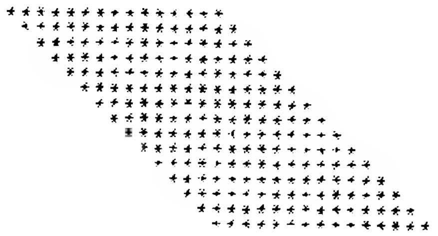
15

D)



12

E)



15

Megoldás

I-A
II-B
III-C
IV-D
V-E

I-D
II-E
III-C
IV-A
V-B

I-C
II-A
III-D
IV-E
V-B

Tesztfeladat 4.

Régi, jól bevált feladattípus a programsorokban eldugott szintaktikai hibák keresése. A tesztlapok változataiban ugyanazzal a négy programsorral „játszottunk", azaz hol az egyiknek hol a másiknak a hibás változatát raktuk be a hibátlan mellé, vagy a hibátlanok közé. A feladatok közt nem közöljük valamennyi változatot, csak három sort adunk meg, ám a megoldásoknál közöljük azoknak a soroknak is a „hibásított" változatát, amelyek itt most hibátlanok (hogy a többes szám indokolt-e, majd kiderül).

A feladat: „A következő három programsort nézd meg figyelmesen! Amelyik sort hibásnak találod, azt jelöld meg a sor melletti négyzetben X-szel, amelyiket hibátlannak, azt 0-val! (Hibás az a BASIC sor, amely elé vagy mögé lehetetlen olyan BASIC sorokat írni, hogy a program a kérdéses soron hibajelzés nélkül végigfusson.)"

```
[ ] 15 FOR I=1 TO 1[illegible]:PRINT I,[illegible](MID$([illegible]STR$(INT[illegible]),2,1));NEXT I
[ ] 20 PRINT "FOR JJ=1 TO 7[illegible]:PRINT JJ:NEXT JJ: [illegible]
[ ] 25 FOR I1=1 TO 3:FOR J1=[illegible] TO -[illegible] STEP -5:[illegible]PRINT I1,J1:NEXT J1:NEXT I1
```

Megoldás

```
|X| 15 FOR I=1 TO 10:PRINT I,VAL(MID$(STR$(INT(I *(I-ABS(I))+1)),2,1));NEXT I
|X| 20 PRINT "FOR JJ6=1 TU 78:PRINT JJ:NEXT JJ6",S$+"5/0"
|O| 25 FOR I1=1 TO 8:FOR J6=6 TO -6 STEP -5:I1=I1+1:PRINT I1,J9:NEXT J6:NEXT I1
```

Azaz az első és a második hibás. Hogy miért, az bizonyára kiderül a további hibás, ill. hibátlan sorok átnézése közben:

```
|O| 15 FOR I=1 TO 10:PRINT I,VAL(MID$(STR$(INT(I *(I+ABS(I))+1)),2,1)):NEXT I
|X| 15 FOR I=1 TO 10:PRINT I,VAL(MID$(STR$(INT(I *(I+ABS(I))+1)),2.1)):NEXT I
|O| 20 PRINT "FOR I=1,34:PRINT I:NEXT I",S$+"9/0";R*R+4*R+3,14159 3.
|O| 20 PRINT "FOR JJ6=1 TU 78:PRINT JJ:NEXT JJ6"
|X| 20 PRINT "FOR I=1,34:PRINT I:NEXT I",S$+"5/0";
|X| 25 FOR I1=1 TO 8:FOR J6=6 TO -6 STEP -5:I1=I1+1:PRINT I1,J9:NEXT I1
|O| 25 FOR I1=1 TO 8:FOR J6=6 TO -6 STEP -5:I1=I1+1:PRINT I1,J9:NEXT J6
```

Tesztfeladat 5.

Ez ismét olyan feladat, amelyből csak egyféle készült, s valamennyi tesztlapon ugyanaz szerepelt.

Feladat: szerinted van-e értelme a következő programocskának?

Ha igen — miért?
Ha nem — miért nem?"

```
10 INPUT X,Y
20 X=X+Y:Y=X-2*Y
30 X=(X+Y)/2:Y=X-Y
40 PRINT X,Y
```

Megoldás Sok értelme nincs, hiszen a program végeredményben az X, Y értékekkel semmit sem csinál, hanem változatlanul visszaírja azokat. Így a 20-as és a 30-as sort akár el is hagyhatnánk! (De legjobb, ha az egész programot elfelejtjük!)

Tesztfeladat 6.

Ez ismét egy elég újszerű, cselesre sikeredett feladattípus. A lényege, hogy a megadott programsorok között oly módon kell „rendet" tenni, hogy azonos betűjellel kell megjelölni azokat a sorokat, amelyeknek lefuttatva teljesen azonos a hatásuk. Persze a dolog nem könnyű, hiszen a sorok mindössze apróságokban különböznek egymástól. A tesztlapokhoz összesen tízféle sor készült. Most ezeket egyszerre

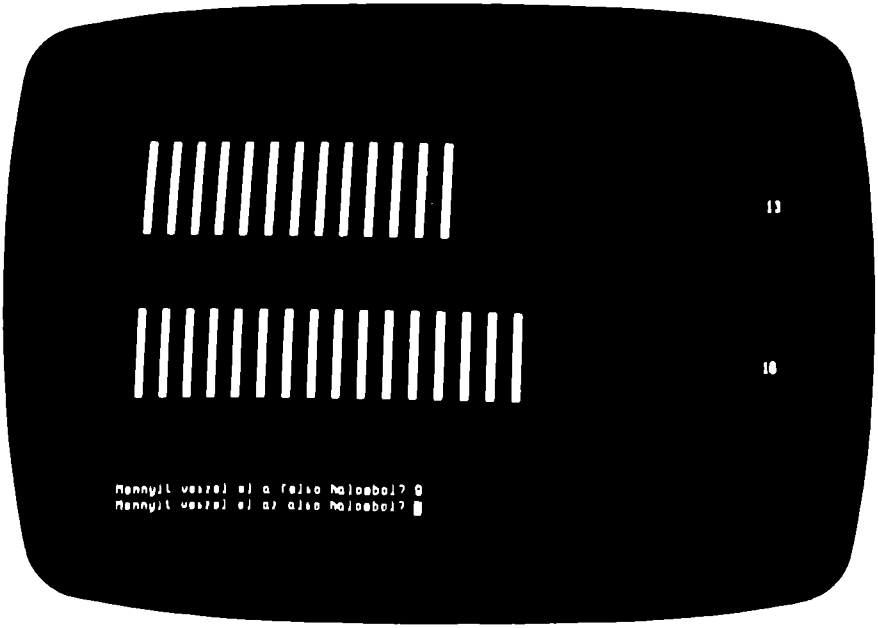
közöljük. Így persze nehezebb a feladat, mint a vetélkedőben, hiszen ott csak 4—4 sor szerepelt egy tesztlapon. (De idő sem sok jutott rá!)

A feladat, hogy a tíz, itt közölt sorból tessék tehát kikeresni és azonos betűjelekkel jelölni azokat a sorokat, amelyeknek teljesen azonos a hatásuk. Ha akad olyan sor is, amelyiknek egyetlen párja sincs, azt is tessék megjelölni valamilyen, a többitől különböző betűvel! (A programok M08X-en és HT1080Z-n működőképesek.)

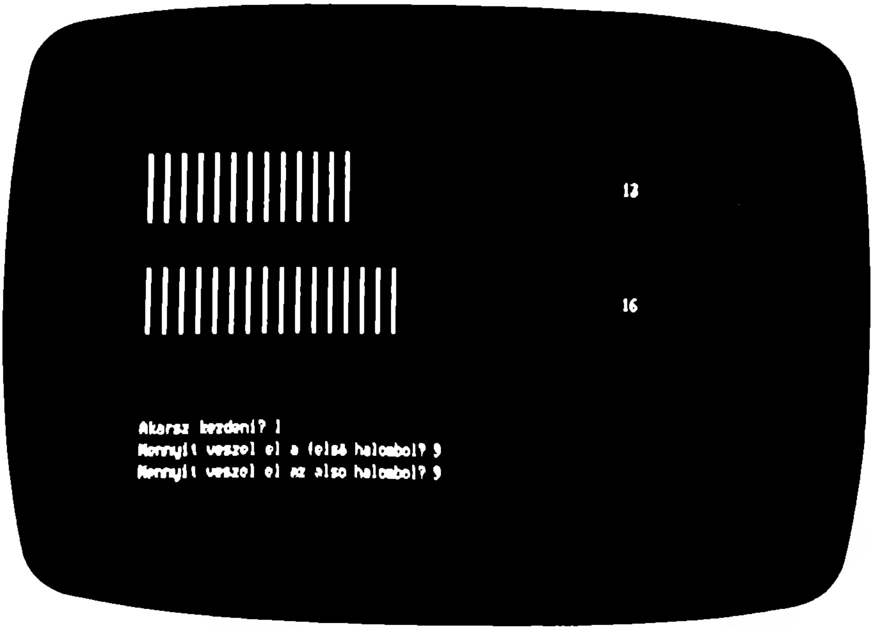
```
|_| 100 IF Y-.5<=X^2+.5 THEN PRINT X*X*X;"##";"#!"
|_| 110 IF X^2>Y-1 THEN PRINT X*X*X;"###";"!" ELSE PRINT,
|_| 120 IF Y>=X^2+1 THEN PRINT; ELSE PRINT X*X*X;"#";"##!"
|_| 130 IF Y-.5<X^5+.5 THEN PRINT X*X*X;"##"+"#!"
|_| 140 IF X^2+1>Y THEN PRINT X*X*X;"###"+"!" ELSE PRINT"";
|_| 150 IF Y-.5<=X^2+.5 THEN PRINT X*X*X;"##;"#!"
|_| 160 IF X^2>Y-1 THEN PRINT X*X*X;"###";"!" ELSE PRINT;
|_| 170 IF Y>X^2-1 THEN PRINT; ELSE PRINT X*X*X;"#";"##!"
|_| 180 IF Y-.5<=X^2+.5 THEN PRINT X*X*X;"##","#!"
|_| 190 IF Y>=X^2+1 THEN PRINT; ELSE PRINT X*X*X+"#";"##!"
```

Megoldás

```
|B| 100 IF Y-.5<=X^2+.5 THEN PRINT X*X*X;"##";"#!"
|C| 110 IF X^2>Y-1 THEN PRINT X*X*X;"###";"!" ELSE PRINT,
|A| 120 IF Y>=X^2+1 THEN PRINT; ELSE PRINT X*X*X;"#";"##!"
|G| 130 IF Y-.5<X^5+.5 THEN PRINT X*Y*X;"##"+"#!"
|A| 140 IF X^2+1>Y THEN PRINT X*X*X;"###"+"!" ELSE PRINT"";
|D| 150 IF Y-.5<=X^2+.5 THEN PRINT X*X*X;"##;"#!"
|A| 160 IF X^2>Y-1 THEN PRINT X*X*X;"###";"!" ELSE PRINT;
|B| 170 IF Y>X^2+1 THEN PRINT; ELSE PRINT X*X*X;"#";"##!"
|E| 180 IF Y-.5<=X^2+.5 THEN PRINT X*X*X;"##","#!"
|F| 190 IF Y>=X^2+1 THEN PRINT; ELSE PRINT X*X*X+"#";"##!"
```

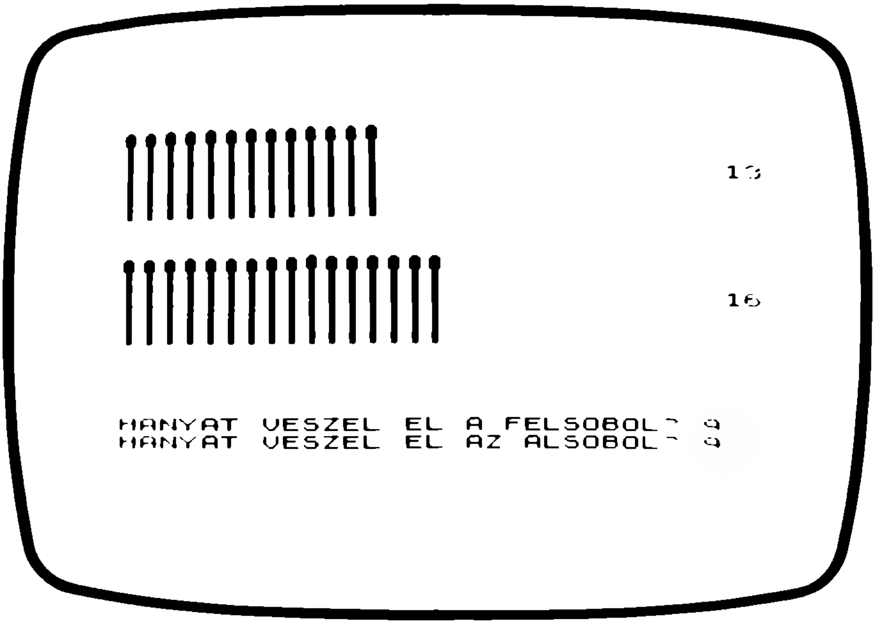


28. képernyőkép
Gyufás játék



29. képernyőkép
Gyufás játék

30. képernyőkép
Gyufás játék



31. képernyőkép
Gyufás játék

A vetélkedőbe benevezettek első feladata egy nagyon rövid játék volt a számítógép ellen. A játék a jól ismert NIM játék egyik változata. A játékosok 10 percnyi felkészülési időt kaptak, s erre az időre megkapták a játékszabályt is. Ezután mindenki a játék kezdetétől számított két percet tölthetett a gép mellett, s ez idő alatt vagy megnyerte a partiját, vagy elveszítette. Olvasóink a játékot lejátszhatják fejben is, de be is írhatják a kész programot gépükbe, hiszen a következőkben közöljük előbb a játékszabályt, majd a négy különböző géphez írt programot.

A játékszabályok

„Két halomban gyufák vannak. A játékot játszó két játékos felváltva vehet el a halmokból a következők szerint:

1. Az egyik halomból vesz el tetszőleges számú gyufaszálat, vagy
2. mindkét halomból elvesz, de ugyanannyit.

A játékot az nyeri, aki az utolsó szál gyufát veszi el."

Idáig az idézett játékszabály. Itt két szám következett, a két halom gyufáinak a száma. Ez más-más volt a selejtező különböző napjain. Itt az általunk közölt programban most ez 13 és 16, de tetszés szerint változtatható a 40-es (Commodore-64-en 30-as, a Sinclair Spectrumon az 50-es) sorban található H1 és H2 értékének átállításával. Bizonyos érték fölé nem mehet a gyufák száma, ui. nem férnek ki a képernyőre.

Ezek után a programok:

```
10 '######## GYUFAS JATEK ########
20 DEFINT H-N
30 MAXIDO=7000
40 CC=63536':H1=13:H2=16:REM ##### KEZDOHALMOK MEGADASA #####
50 DATA 0,0,1,2,3,5,4,7,6,10,8,13,9,15,11,18,12,20,14,23,16,26,17,28,19,31,21,34
,22,36
60 DIM H1(15),H2(15)
70 FOR I=1 TO 15 :READ H1(I),H2(I):NEXT I
80 REM ##### JATEK KEZDETE #####
90 GOSUB 1000
100 SOR=20:OSZ=0:GOSUB 1400:TIP=0:GOSUB 1200
110 Q$=INKEY$:PRINT "Akarsz kezdeni? ";
120 IDO=IDO+1:Q$=INKEY$:IF Q$="" THEN 120
125 RANDOMIZE IDO
130 PRINT Q$
140 IF Q$="n" OR Q$="N" THEN 500
200 REM ##### A JATEKOS LEPESE #####
210 OSZ=35:SOR=20:GOSUB 1400:PRINT"                    ";:OSZ=0:GOSUB 1400
220 Q$=INKEY$:PRINT "Mennyit veszel el a felso halombol? ";
230 GOSUB 800
240 IF HAP=1 THEN 200
250 V1=VAL(Q$):IF V1>H1 THEN 200
260 OSZ=35:SOR=21:GOSUB 1400:PRINT"                    ";:OSZ=0:GOSUB 1400
270 Q$=INKEY$:PRINT "Mennyit veszel el az also halombol? ";
280 GOSUB 800
290 IF HAP=1 THEN 260
300 V2=VAL(Q$):IF V2>H2 THEN 260
310 IF (V1<>V2 AND V1<>0 AND V2<>0) OR V1+V2=0 THEN PRINT"SZABALYTALAN ELVETEL
!!!":PRINT"Ird ujra be, hogy mit veszel el!":GOTO 200
320 H1=H1-V1:H2=H2-V2
330 GOSUB 1000
340 SOR=20:GOSUB 1400
350 IF H1+H2=0 THEN PRINT"GRATULALOK, TE NYERTEL!!!":END
360 PRINT"Ha megnezted, nyomj meg egy szurke gombot!"
370 A$=INKEY$:IDO=IDO+1:IF IDO>MAXIDO THEN 1600 ELSE IF A$="" THEN 370
500 REM ##### A GEP LEPESE #####
```

```
 510 E1=-(RND(.5):E2=-(RND(.5):IF E1+E2=0 THEN 510
 520 FOR I=1 TO 15
 530 IF H1=H1(I) AND H2(I)<H2 THEN E1=0:E2=H2-H2(I):GOTO 600
 540 IF H1=H2(I) AND H1(I)<H2 THEN E1=0:E2=H2-H1(I): GOTO 600
 550 IF H2=H1(I) AND H2(I)<H1 THEN E1=H1-H2(I):E2=0:GOTO 600
 560 IF H2=H2(I) AND H1(I)<H1 THEN E1=H1-H1(I):E2=0:GOTO 600
 570 IF H1-H1(I)=H2-H2(I) AND H1>H1(I) THEN E1=H1-H1(I):E2=E1:GOTO 600
 580 IF H1-H2(I)=H2-H1(I) AND H1>H2(I) THEN E1=H1-H2(I):E2=E1:GOTO 600
 590 NEXT I
 600 H1=H1-E1:H2=H2-E2:GOSUB 1000:SOR=19:GOSUB 1400
 610 PRINT"A felsobol";E1;"-t, az alsobol";E2;"-t vettem el."
 620 IF H1+H2=0 THEN PRINT:PRINT"EN NYERTEM!!!!!":END
 630 GOTO 200
 800 REM ##### SZAM BEKERDEZES #####
 810 KAP=0
 820 Q$=INKEY$:IDO=IDO+1:IF IDO>MAXIDO THEN 1600 ELSE IF Q$="" THEN 820
 830 PRINT Q$;
 840 Q1$=INKEY$:IDO=IDO+1:IF IDO>MAXIDO THEN 1600 ELSE IF Q1$="" THEN 840
 850 PRINT Q1$;:IF ASC(Q1$)<>13 THEN Q$=Q$+Q1$:WHILE INKEY$="":IDO=IDO+1:IF IDO>M
AXIDO THEN 1600 ELSE WEND:PRINT
 860 IF LEN(Q$)>2 OR Q$<"0" OR Q$>"99" THEN KAP=1
 870 IF LEN(Q$)=2 THEN IF Q1$<"0" OR Q1$>"9" THEN KAP=1
 880 RETURN
1000 REM ##### MEGJELENITES #####
1010 PRINT CHR$(17);
1020 FOR JJ=1 TO 5:POKE CC+JJ*80+3*H1,128:NEXT JJ
1030 FOR II=0 TO H1-1:FOR JJ=1 TO 5
1040 POKE CC+JJ*80+3*II,128:POKE CC+JJ*80+3*II+1,144:NEXT JJ,II
1050 POKE CC+316,129:POKE CC+318,48+H1 MOD 10:IF H1>9 THEN POKE CC+317,48+INT(H1
/10)
1060 FOR JJ=10 TO 14:POKE CC+JJ*80+3*H2,128:NEXT JJ
1070 FOR II=0 TO H2-1:FOR JJ=10 TO 14
1080 POKE CC+JJ*80+II*3,128:POKE CC+JJ*80+II*3+1,144:NEXT JJ,II
1090 POKE CC+1036,129:POKE CC+1038,48+H2 MOD 10:IF H2>9 THEN POKE CC+1037,48+INT
(H2/10)
1100 RETURN
1190 REM ##############################
1200 REM ##### SUBR. KURZOR TIPUS #####
1210 REM ##### TIP=A kurzor tipusa. ###
1220 REM ##### 0: Villogo, forditott ##
1230 REM ##### 1: Villogo, alahuzott ##
1240 REM ##### 2: Forditott ###########
1250 REM ##### 3: Alahuzott ###########
1260 REM ##############################
1270 TIP=INT(ABS(TIP)):IF TIP<0 OR TIP>3 THEN RETURN
1280 PRINT CHR$(27);"[";CHR$(TIP+48);"q";
1290 RETURN
1390 REM #############################
1400 REM #### SUBR. KURZOR POZ. #####
1410 REM #### SOR=Sorszam (0-24) ####
1420 REM #### OSZ=Oszlopszam (0-79) #
1430 REM #### Hasznalt valtozok: ####
1440 REM #### S1, S2, O1, O2, AAA$. #
1450 REM #############################
1460 SOR=INT(ABS(SOR)):OSZ=INT(ABS(OSZ))
1470 IF SOR<0 OR SOR>24 OR OSZ<0 OR OSZ>79 THEN RETURN
1480 S1=INT(SOR/10)+48:S2=SOR MOD 10+48:O1=INT(OSZ/10)+48:O2=OSZ MOD 10+48
1490 AAA$="["+CHR$(S1)+CHR$(S2)+CHR$(O1)+CHR$(O2)+"f"
1500 PRINT CHR$(27);AAA$;
1510 RETURN
1600 REM ###### LEJART AZ IDO ########
1610 PRINT CHR$(17):PRINT:PRINT "Sajnos lejart az idod, igy vesztettel.":END
```

```
10 '########### GYUFAS JATéK ##############
20 DEFINT E,H-N,V:DEFSTR G,Q
30 MAXIDO=4200
40 H1=13:H2=16:REM ##### KEZDöHALMOK MEGADASA #####
45 FOR I=1 TO 31:G=G+CHR¤(170)+" ":NEXT I
50 DATA 0,0,1,2,3,5,4,7,6,10,8,13,9,15,11,18,12,20,14,23,16,26,17,28,19,31
60 DIM H1(13),H2(13)
70 FOR I=1 TO 13:READ H1(I),H2(I):NEXT I
80 '######## A JATéK KEZDETE #########
90 GOSUB 1000
100 Q=INKEY¤:PRINT é768,"Akarsz kezdeni? ";
110 IDO=IDO+1:Q=INKEY¤:IF Q="" THEN 110
120 PRINT Q
130 IF Q="N" THEN 500
200 '######### A JáTéKOS LéPéSE ###########
210 PRINT é867,CHR¤(202);
220 Q=INKEY¤:PRINT é832,"Mennyit veszel el a felso halombol? ";
230 GOSUB 800
240 IF KAP=1 THEN 200
250 V1=VAL(Q):IF V1>H1 THEN PRINT é960,"Ennyit nem vehetsz el!
                    ";:GOTO 200
255 PRINT é960,CHR¤(252);
260 PRINT é931,CHR¤(202);
270 Q=INKEY¤:PRINT é896,"Mennyit veszel el az also halombol? ";
280 GOSUB 800
290 IF KAP=1 THEN 260
300 V2=VAL(Q):IF V2>H2 THEN PRINT é960,"Ennyit nem vehetsz el'
                    ";:GOTO 260
310 IF (V1<>V2 AND V1<>0 AND V2<>0) OR V1+V2=0 THEN PRINT é964, "Szabálytalan el
vétel!!' Ird be ujra, hogy mit veszel el!";:PRINT é896,CHR¤(252);:GOTO 200
320 H1=H1-V1:H2=H2-V2
330 GOSUB 1000
340 IF H1=0 AND H2=0 THEN PRINT é768,"GRATULáLOK, TE NYERTéL!":END
350 Q=INKEY¤:PRINT é960,"Ha megnezted, nyomj meg egy gombot!
  ";
360 Q=INKEY¤:IDO=IDO+1:IF IDO>MAXIDO THEN 1600 ELSE IF Q="" THEN 360
370 PRINT é960,CHR¤(252);
500 '######### A GéP LéPéSE ###########
510 E1=RND(2)-1:E2=RND(2)-1:IF E1+E2=0 THEN 510
520 FOR I=1 TO 13
530 IF H1=H1(I) AND H2(I)<H2 THEN E1=0:E2=H2-H2(I):GOTO 600
540 IF H1=H2(I) AND H1(I)<H2 THEN E1=0:E2=H2-H1(I):GOTO 600
550 IF H2=H1(I) AND H2(I)<H1 THEN E1=H1-H2(I):E2=0:GOTO 600
560 IF H2=H2(I) AND H1(I)<H1 THEN E1=H1-H1(I):E2=0:GOTO 600
570 IF H1-H1(I)=H2-H2(I) AND H1>H1(I) THEN E1=H1-H1(I):E2=E1:GOTO 600
580 IF H1-H2(I)=H2-H1(I) AND H1>H2(I) THEN E1=H1-H2(I):E2=E1:GOTO 600
590 NEXT I
600 H1=H1-E1:H2=H2-E2:GOSUB 1000
610 PRINT é768,"A felsobol";E1;"-t, az alsobol";E2;"-t vettem el."
620 IF H1+H2=0 THEN PRINT é832,"é N    N Y E R T E M ' ' !":END
630 GOTO 200
800 REM ######## SZáM BEKéRDEZéS ###########
810 KAP=0
820 Q=INKEY¤:IDO=IDO+1:IF IDO>MAXIDO THEN 1600 ELSE IF Q="" THEN 820
830 PRINT Q;
840 Q1=INKEY¤:IDO=IDO+1:IF IDO>MAXIDO THEN 1600 ELSE IF Q1="" THEN 840
850 PRINT Q1;:IF ASC(Q1)=13 THEN 870
860 QQ=INKEY¤:IDO=IDO+1:IF IDO>MAXIDO THEN 1600 ELSE IF QQ="" THEN 860 ELSE Q=Q+
Q1
870 IF LEN(Q)>2 OR Q<"0" OR Q>"99" THEN KAP=1
880 IF LEN(Q)=2 THEN IF Q1<"0" OR Q1>"9" THEN KAP=1
```

```
890 IF KAP=1 THEN PRINT £960,"Nem idetartozo karakter!
    ";
°CC RETURN
1000 '########### MEGJELENITéS ############
1010 CLS
1020 FOR II=1 TO 3:PRINT LEFT¤(G,2*H1):NEXT II
1030 PRINT:PRINT:FOR II=1 TO 3:PRINT LEFT¤(G,2*H2):NEXT II
1040 PRINT £125,RIGHT¤(STR¤(H1),2);
1050 PRINT £445,RIGHT¤(STR¤(H2),2);
1060 RETURN
1600 '######## LEJART AZ IDÖ ##########
1610 CLS:PRINT:PRINT:PRINT "Sajnos lejárt az idöd, igy vesztettél.":PRINT:END
```

Megjegyzés

Azoknál a HT1080Z gépre készült programoknál, amelyekben ékzetes betűk is szerepelnek, kénytelenek voltunk az ún. svéd karakterkészletet használni a nyomtatás során. Ezekben a programokban tehát dollárjel helyett ¤ , és mint ahogy az lenni szokott a @ jel helyett É szerepel.

```
10 REM ########## GYUFAS JATEK #########
20 MI=8000
30 H1=13:H2=16:REM ##### KEZDOHALMOK ####
40 DATA 0,0,1,2,3,5,4,7,6,10,8,13,9,15,11,18,12,20,14,23,16,26,17,28,19,31,21,34
41 DATA 22,36
50 DIM H1(15),H2(15)
60 FOR I=1 TO 15:READ H1(I),H2(I):NEXT I
65 SP$="                                         "
70 F$=" ●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●"
75 G$=" |||||||||||||||||||||||||||||||||||||||"
80 REM ######## A JATEK KEZDETE ########
90 GOSUB 1000
100 POKE198,0:PRINT "AKARSZ KEZDENI? ";
110 GET Q$:ID=ID+1:IF Q$="" THEN 110
120 PRINT Q$:PRINT
130 IF Q$="N" THEN 500
200 REM ######## A JATEKOS LEPESE ######
210 PRINT SP$:PRINT "]";
220 POKE 198,0:PRINT "MENNYIT VESZEL EL A FELSO HALOMBOL? ";
230 GOSUB 800
240 IF KAP=1 THEN PRINT "]";:GOTO 200
250 V1=VAL(Q$):IF V1>H1 THEN PRINT "]";:GOTO 200
260 PRINT SP$:PRINT"]";
270 POKE 198,0 PRINT "MENNYIT VESZEL EL AZ ALSO HALOMBOL? ";
280 GOSUB 800
290 IF KAP=1 THEN PRINT"]";:GOTO 260
300 V2=VAL(Q$):IF V2>H2 THEN PRINT"]";:GOTO 260
310 IF (V1<>V2 AND V1<>0 AND V2<>0) OR V1+V2=0 THEN 700
320 H1=H1-V1:H2=H2-V2
330 GOSUB 1000
340 IF H1+H2=0 THEN PRINT "]]]GRATULALOK,TE NYERTEL.":PRINT:END
350 PRINT "███HA MEGNEZTED NYOMJ MEG EGY GOMBOT!";:POKE 198,0
360 ID=ID+1:IF ID>MI THEN 1600
370 GET Q$:IF Q$="" THEN 360
3?0 PRINT    "████████████████████████████████████████"+SP$;
500 REM ######### A GEP LEPESE #########
510 E1=-(RND(1)<.5):E2=-(RND(1)<.5):IF E1+E2=0 THEN 510
520 FOR I=1 TO 15
530 IF H1=H1(I) AND H2(I)<H2 THEN E1=0:E2=H2-H2(I):GOTO 600
540 IF H1=H2(I) AND H1(I)<H2 THEN E1=0:E2=H2-H1(I):GOTO 600
550 IF H2=H1(I) AND H2(I)<H1 THEN E2=0:E1=H1-H2(I):GOTO 600
560 IF H2=H2(I) AND H1(I)<H1 THEN E2=0:E1=H1-H1(I):GOTO 600
570 IF H1-H1(I)=H2-H2(I) AND H1>H1(I) THEN E1=H1-H1(I):E2=E1:GOTO 600
580 IF H1-H2(I)=H2-H1(I) AND H1>H2(I) THEN E1=H1-H2(I):E2=E1:GOTO 600
590 NEXT I
600 H1=H1-E1:H2=H2-E2:GOSUB 1000
```

```
610 PRINT "FELULROL";STR$(E1);"-T, ALULROL",STR$(E2);"-T VETTEM EL.N"
620 IF H1+H2=0 THEN PRINT "EN NYERTEM!!!":END
630 GOTO 200
700 PRINT "SZABALYTALAN ELVETEL! IRD BE UJRA!";
710 PRINT "▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒[]";
720 GOTO 200
800 REM ######## SZAM BEKERDEZES #######
810 KAP=0
820 GET Q$:ID=ID+1:IF ID>MI THEN 1600
825 IF Q$="" THEN 820
826 IF ASC(Q$)=13 THEN 820
830 PRINTQ$;
840 GET Q1$:ID=ID+1:IF ID>MI THEN 1600
845 IF Q1$="" THEN 840
850 IF ASC(Q1$)=13 THEN 870
860 PRINT Q1$;:Q$=Q$+Q1$:FOR I=ID TO MI:GET QQ$:IF QQ$="" THEN NEXT I:GOTO 1600
870 IF Q$<"0" OR Q$>"99" THEN KAP=1
880 IF LEN(Q$)=2 THEN IF Q1$<"0" OR Q1$>"9" THEN KAP=1
890 PRINT:RETURN
1000 REM ######## MEGJELENITES #########
1005 PRINT "▒▒▒";LEFT$(F$,H1+1)
1010 PRINT LEFT$(G$,H1+1):PRINT LEFT$(G$,H1+1);TAB(37);RIGHT$(STR$(H1),2)
1015 PRINT LEFT$(G$,H1+1):PRINT LEFT$(G$,H1+1)
1020 PRINT "▒▒";LEFT$(F$,H2+1)
1025 PRINT LEFT$(G$,H2+1):PRINT LEFT$(G$,H2+1);TAB(37);RIGHT$(STR$(H2),2)
1030 PRINT LEFT$(G$,H2+1):PRINT LEFT$(G$,H2+1)
1035 PRINT "▒▒▒▒▒";:RETURN
1600 REM ######## LEJART AZ IDO ########
1610 PRINT "▒▒▒▒";"SAJNOS LEJART AZ IDOD, IGY VESZTETTEL.":PRINT:END
```

```
 10 REM ##### GYUFAS JATEK ####
 20 DATA 0,24,24,24,60,24,60,24,60,24,60,24,60,24,24,24
 30 FOR I=0 TO 7: READ H,G: POKE USR "H"+I,H: POKE USR "G"+I,G:
NEXT I: REM ### KARAKTER DEF. #
 40 LET MI=4500: LET ID=0
 50 LET H1=13: LET H2=16: REM    ### KEZDOHALMOK ###
 60 LET F$="▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮": LET A$="|||||||||
||||||||||||||||||||||"
 65 DIM E(13): DIM M(13)
 70 DATA 0,0,1,2,3,5,4,7,6,10,8,13,9,15,11,18,12,20,14,23,16,26
,17,28,19,31
 80 FOR I=1 TO 13: READ E(I),M(I): NEXT I
 90 REM ### A JATEK KEZDETE ###
100 GO SUB 999
110 LET Q$=INKEY$: PRINT AT 17,0;"AKARSZ KEZDENI? ";
120 LET ID=ID+1: LET Q$=INKEY$: IF Q$="" THEN  GO TO 120
130 PRINT Q$: IF Q$="N" OR Q$="n" THEN  GO TO 500
140 REM ### A JATEKOS LEPESE ##
150 PRINT AT 17,28;"    ";
160 LET Q$=INKEY$: PRINT AT 17,0;"HANYAT VESZEL EL A FELSOBOL?
";
170 LET K=0: GO SUB 800
180 LET V1=VAL Q$: IF V1>H1 THEN  GO TO 150
185 PRINT AT 20,0;"                                ": PRINT "
                              ";
190 PRINT AT 18,28;"    ";
200 LET Q$=INKEY$: PRINT AT 18,0;"HANYAT VESZEL EL AZ ALSOBOL?
";
210 LET K=1: GO SUB 800
220 LET V2=VAL Q$: IF V2>H2 THEN  GO TO 190
230 IF (V1<>V2 AND V1<>0 AND V2<>0) OR V1+V2=0 THEN  PRINT AT 2
0,0;"SZABALYTALAN ELVETEL!          ": PRINT "IRD UJRA BE,HOGY MI
```

```
T VESZEL EL!";: GO TO 150
 240 LET H1=H1-V1: LET H2=H2-V2
 250 GO SUB 999
 260 IF H1+H2=0 THEN  PRINT AT 17,0;"GRATULALOK, TE NYERTEL!": S
TOP
 270 PRINT AT 17,0;"HA MEGNEZTED, NYOMJ MEG EGY                G
OMBOT!": LET Q$=INKEY$
 280 LET ID=ID+1: IF ID>MI THEN  GO TO 900
 290 LET Q$=INKEY$: IF Q$="" THEN  GO TO 280
 300 PRINT AT 17,0;"
                  ";
 500 REM #### A GEP LEPESE #####
 510 LET E1=RND<.5: LET E2=RND<.5: IF E1+E2=0 THEN  GO TO 510
 520 FOR I=1 TO 13
 530 IF H1=E(I) AND M(I)<H2 THEN  LET E1=0: LET E2=H2-M(I): GO T
O 600
 540 IF H1=M(I) AND E(I)<H2 THEN  LET E1=0: LET E2=H2-E(I): GO T
O 600
 550 IF H2=E(I) AND M(I)<H1 THEN  LET E2=0: LET E1=H1-M(I): GO T
O 600
 560 IF H2=M(I) AND E(I)<H1 THEN  LET E2=0: LET E1=H1-E(I): GO T
O 600
 570 IF H1-E(I)=H2-M(I) AND H1>E(I) THEN  LET E1=H1-E(I): LET E2
=E1: GO TO 600
 580 IF H1-M(I)=H2-E(I) AND H1>M(I) THEN  LET E1=H1-M(I): LET E2
=E1: GO TO 600
 590 NEXT I
 600 LET H1=H1-E1: LET H2=H2-E2: GO SUB 999
 610 PRINT AT 20,0;"A FELSOBOL ";E1;"-T, AZ ALSOBOL": PRINT E2;"
-T VETTEM EL.";
 620 IF H1=0 AND H2=0 THEN  PRINT AT 17,0;"EN NYERTEM!!!": STOP
 630 GO TO 150
 800 REM ### SZAM BEKERDEZES ###
 810 FOR I=1 TO 50: NEXT I: LET B$=INKEY$
 820 LET ID=ID+1: IF ID>MI THEN  GO TO 900
 830 LET Q$=INKEY$: IF Q$="" THEN  GO TO 820
 840 PRINT AT 17+K,29;Q$;
 845 FOR I=1 TO 50: NEXT I: LET B$=INKEY$
 850 LET ID=ID+1: IF ID>MI THEN  GO TO 900
 860 LET W$=INKEY$: IF W$="" THEN  GO TO 850
 870 PRINT AT 17+K,30;W$;: IF CODE W$=13 THEN  GO TO 890
 872 FOR I=1 TO 50: NEXT I: LET B$=INKEY$
 875 LET ID=ID+1: IF ID>MI THEN  GO TO 900
 880 LET B$=INKEY$: IF B$="" THEN  GO TO 875
 885 LET Q$=Q$+W$
 890 IF Q$<"0" OR Q$>"99" THEN  GO TO 150+K*40
 892 IF LEN Q$=2 THEN  IF Q$(2)<"0" OR Q$(2)>"9" THEN  GO TO 150
+K*40
 895 RETURN
 900 REM ##### LEJART AZ IDO ###
 910 PRINT AT 20,0;"SAJNOS LEJART AZ IDOD, IGY TE              V
ESZTETTEL!          ";: STOP
 999 REM ##### MEGJELENITES ####
1000 CLS : PRINT AT 1,0;F$( TO H1);
1010 FOR I=2 TO 5: PRINT AT I,0;A$( TO H1);: NEXT I
1020 PRINT AT 8,0;F$( TO H2);
1030 FOR I=9 TO 12: PRINT AT I,0;A$( TO H2);: NEXT I
1040 PRINT AT 3,30;(" "+STR$ H1)((H1>9)+1 TO );
1050 PRINT AT 10,30;(" "+STR$ H2)((H2>9)+1 TO );
1060 RETURN
```

A program felépítése, működése

A program a kezdő halmok megadásával indul. A H1 és H2 változókba tetszés szerint írhatunk számokat. Ajánljuk, hogy 25 alatti számok legyenek ezek!

Az ezután következő DATA utasítások a számítógép nyerő stratégiájának lelkét képezik, ui. itt vannak azok a számpárok, amelyek a nyeréshez szükségesek. Ezeket a program mindig beolvassa a H1 és a H2 vektorokba.

Miután a kezdő értékek beolvasása megtörtént, a program meghívja az 1000-es rutint, amely kirajzolja a képernyőre a megfelelő számú gyufát és mellé írja a gyufák számát is. Ezután megkérdezi a játékostól, hogy akar-e kezdeni, és a választól függően a 200-as — a Sinclair Spectrumon a 140-es — (a játékos lépése), vagy az 500-as (a gép lépése) sorokon folytatódik a program.

A játékos lépése

A gép megkérdezi a játékostól, hogy mennyit vesz el a felső, ill, az alsó halomból. A választ a 800-as rutin fogadja, amely időt is mér (ettől lett ilyen hosszú). Amennyiben a beadott szám negatív vagy nem egész, akkor a KAP nevű kapcsolót 1-re állítja. A 800-as rutinból visszatérve, ha a KAP értéke 1, vagy a beírt szám nagyobb a halom elemszámánál, akkor a program újra kérdez. (A Sinclair Spectrumon kapcsolóra nincs szükség, helyette kiszámított GOTO utasítást használtunk.) Miután mindkét számot rendben találta és elfogadta, még ellenőrzi, hogy a szabályoknak megfelelően vettük-e el a gyufaszálakat, azaz megvizsgálja, hogy nem két nullát írtunk-e be, ill. a két halomból elvett gyufák száma egyezik-e. Ez utóbbira persze nincs szükség, ha egyik halomból semmit sem veszünk el.

Az időmérés során, ha a felhasznált idő túllépi a játék keretidejét (2 perc), akkor a program az 1600-as — a Sinclair Spectrumon 900-as — sorra megy, és nemes egyszerűséggel közli, hogy a játékos vesztett.

Amennyiben mindkét érték rendben van, és idő is van még a további játékra, akkor a program beállítja a H1 és H2 változók új értékét, és meghívja az 1000-es megjelenítőrutint. A képernyőn megjelenik a játékos lépése utáni helyzet, majd miután a megjelenítőrutinból visszatért a program, megvizsgálja, vajon van-e még egyáltalán gyufa a képernyőn. Ha nincs, gratulál a játékosnak a nyeréshez. Ha van, akkor viszont egy tetszőleges billentyű megnyomását kéri.

Ha ez megtörtént, továbblép a gép, és a saját lépését számolja ki a DATA utasítások segítségével.

A gép lépése

Először E1-nek és E2-nek véletlenszerűen generál 0 vagy 1 értéket. Ezekben tárolja a gép saját — éppen aktuális — lépését. Persze vigyáz arra, hogy ne legyen mindkettő nulla! Erre az egészre azért van szükség, hogy a gép vesztő állásban is tudjon valamit lépni.

Ezután következik a gép „gondolkodása" saját lépésén. Azaz a program végigvizsgálja, hogy tud-e lépni valamelyik nyerő halompárra. Ha igen, akkor jó, ha nem, akkor megmarad az E1 és az E2 véletlenszerűen választott értéke.

Most a gép beállítja a H1 és a H2 változók értékét, majd elmegy a megjelnítőrutinra, ahonnan a gyufák kirajzolása után visszajön és kiírja a saját lépését. Megvizsgálja, hogy van-e még gyufa a képernyőn, s ha nincs, nagyképűen jelzi, hogy nyert. Ha nem, akkor visszatér a „játékos lépése" részre.

Néhány szót a *programok változatairól*, elsősorban az 1000-es megjelenítőrutinról. Érthető, hogy a négy program legtöbb eltérését ez a rutin tartalmazza, hiszen a megjelenítés. a képernyőkezelés tér el leginkább a gépeken. Az M08X-en a képernyő-

memória megfelelő részére a megjelenítés az inverzre váltó karakterek elhelyezésével, a másik három gépen egyszerű PRINT utasítással történik. A kinyomtatandó sorokat a program elején definiáljuk, mégpedig a Commodore-64-en saját grafikus, a HT1080Z-n saját kvázigrafikus, a Sinclair Spectrumon definiált grafikus karakterek segítségével.

Ugyancsak nagy különbséget jelent a négy program között a gépek legkülönbözőbb módokon megoldott kurzormozgatási módszere. A Commodore-64 és a Sinclair Spectrum BASIC-jéből hiányzó ELSE, és a Commodore-64-nél az INKEY$ helyett használandó GET is okoz eltéréseket. Legvégül magyarázatként még annyit, hogy a Commodore-64 programjában használt POKE 198,0 — hasznos információ — a billentyűzet pufferjének kiürítését végzi.

Ha valakinek valóban kedve támad a fenti programok közül épp a Sinclair Spectrumra készített változatot beírni gépébe, annak a következőket ajánljuk figyelmébe.

Először is be kell írnia a grafikus karaktereket definiáló programrészt (l. a 20—30 sorokat). Utána ezt le kell futtatni, majd ezután lehet beírni a program további részét. Ha grafikus üzemmódban leütjük az F betűt, ez a képernyőn egy gyufafejet jelenít meg, míg a G betű a gyufa szárának részét.

Megoldás

A nyerő halmok kiolvashatók a programokban található DATA utasításokból. Ezek: 1-2; 3-5; 4-7; 6-10; 8-13; 9-15; 11-18; 12-20; 14-23; 16-26; 17-28; 19-31.

Ezeket egyszerű visszafelé okoskodással „könnyen" meg lehet találni. Néhány versenyzőnek ez sikerült is!

Érdekesség! A nyerő halompároknak van egy általános képlete is. Éspedig:

$$\left(\left[\frac{\sqrt{5}+1}{2}\cdot n\right], \left[\frac{\sqrt{5}+3}{2}\cdot n\right]\right)$$

ahol n természetes szám, a [ ] zárójelpár az egészrész-képzést jelenti.

Vigaszág

Azok az érdeklődők, akik valamiért nem neveztek be a vetélkedő selejtezőjére, vagy onnan nem jutottak tovább, még egy lehetőséget kaptak. Nevezhettek a BNV-n, és itt 15 percet tölthettek egy M08X gép mellett. E 15 perc alatt háromféle feladattal kellett megbirkózniuk, s idejük leteltével e három feladat alapján kaptak egy bizonyos pontszámot.

E vigaszágról a három nap több mint 150 játékosa közül négyen jutottak a vetélkedő döntőjébe.

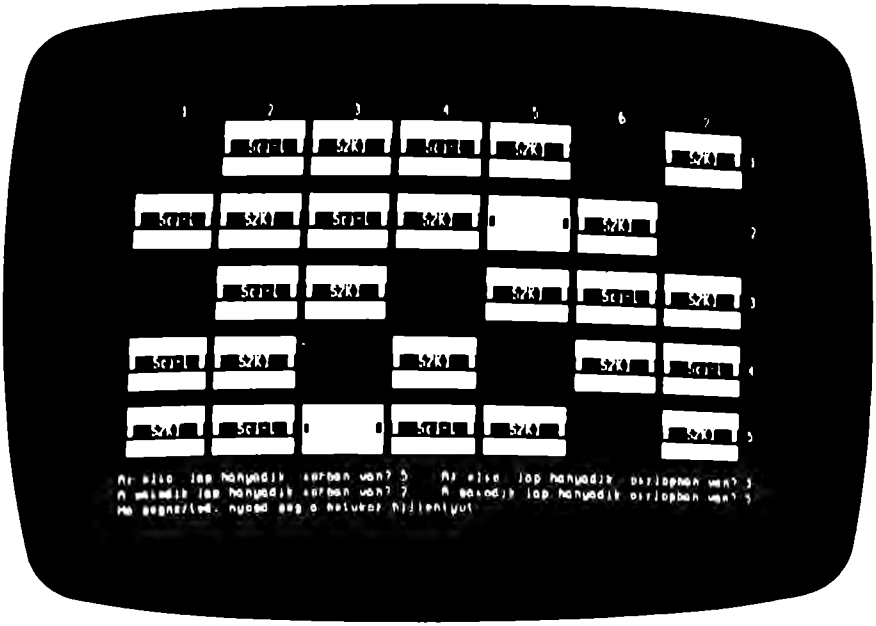
A 15 perces játékidő első feladata egy közkedvelt játék számítógépes változata volt.

Erre a játékra 8 percet kaptak a játékosok, s előtte írásban tanulmányozhatták a játékszabályt, hogy ne kelljen idejüket a szabályolvasással tölteni. Ismerkedjünk meg először a játékszabályokkal, amelyeket a versenyzők is olvashattak.

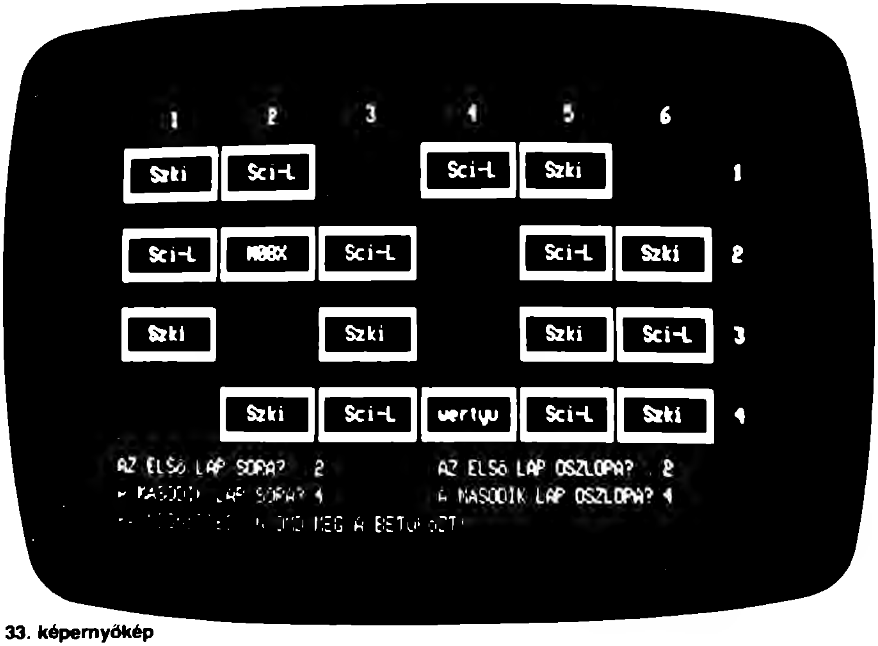
Játékszabályok

,,Bizonyára ismerős ez a játék, amelyben kártyák fekszenek az asztalon, s egymás után két kártyát lehet megfordítani. Ha a két kártyán levő szöveg vagy figura azonos, úgy ezt a két kártyát, azaz ezt a kártyapárt ki lehet emelni, amiért valamennyi pont kapható. Ha nem, akkor a két kártyát vissza kell fordítani.

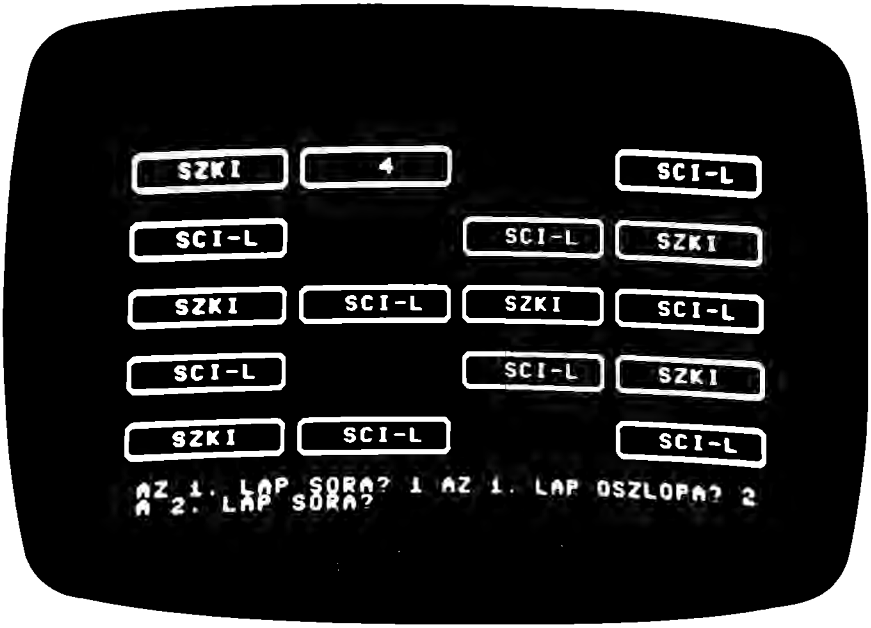
Játékunkban a képernyőn 35 kártyalap jelenik meg.



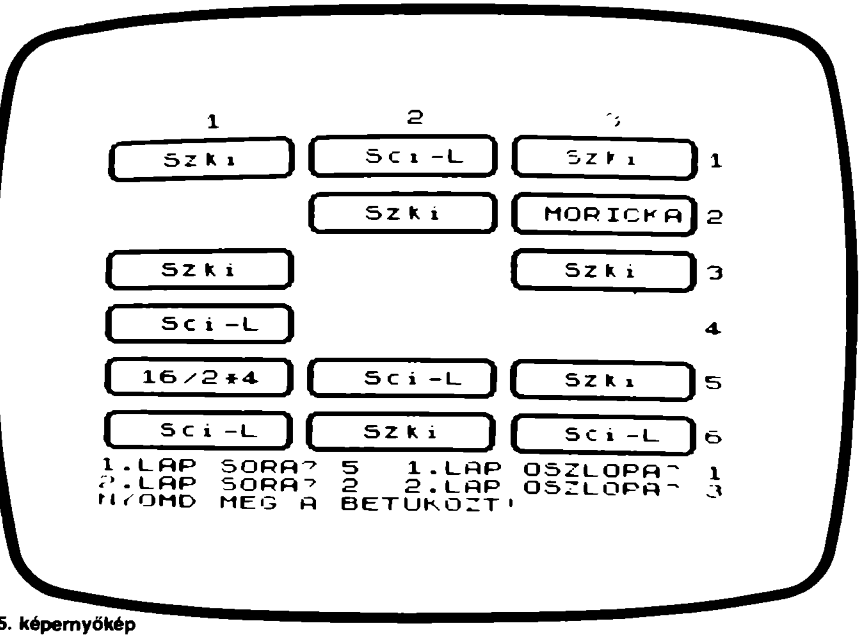
32. képernyőkép
Memory



33. képernyőkép
Memory



34. képernyőkép
Memory



35. képernyőkép
Memory

A kártyalapok hátoldalán SzKI, ill. SCI-L feliratokat láthatsz. A kártyasorok és -oszlopok oldalán, ill. tetején sorszámokat láthatsz. A gép kérdésére, hogy hányadik sorban és oszlopban van az általad éppen kiválasztott kártya, ezeknek a számoknak alapján kell megjelölnöd, hogy melyik lapot akarod megnézni. A megnevezett kártya „megfordul", azaz megjelenik az azon levő szöveg vagy szám. Ezután újabb lapot kell megjelölnöd, s ez is megfordul. Ha azonos a két lap, úgy a gép közli ezt veled, és természetesen ez a két lap ezután eltűnik a képernyőről. Ha a két lap nem azonos, úgy ezt is közli a gép, és a lapokat „visszafordítja". Ezután újabb lapok megnevezését kéri.

A 35 lapban levő párok nem azonos értékűek.

További szabályok:

1. Azok a párok, amelyeken szöveg van, kétfélék lehetnek:

a) értelmes szöveg van a lapokon — ebben az esetben a pár 100 pontot ér;

b) nem értelmes betűcsoport van a lapokon — ez 150 pontot ér.

Megjegyzés: kis- és nagybetűk nem párosíthatók! Tehát a „bnv" és a „BNV" felirat nem pár. A „bnv" párja csak „bnv" lehet!

2. Vannak lapok, amelyeken matematikai kifejezések szerepelnek. Itt azonban a párok alakilag nem azonosak, de matematikailag igen.

Tehát például 2×6/4 feliratú kártya párja egy olyan kártya lehet, amelyen a 3 szerepel. Az ilyen párok megtalálása nehezebb, de többet is ér, 250 pontot!

3. A lapok száma páratlan! Van tehát egy olyan lap, amelynek nincs párja. Ez a legértékesebb! Ha tudod, hogy melyik ez a páratlan lap, s ezt ki akarod venni, a következőképpen járj el.

Add meg a gépnek az adott lap sor- és oszlopszámát, ekkor a lap megfordul. Majd a gép újabb kérdésére, hogy most milyen lapot akarsz megfordítani, ismét ennek a lapnak a koordinátáit add meg. Erre a gép vagy kiírja, hogy valóban ez volt az

egyedülálló lap, ezt leveszi a képernyőről és jóváír neked 300 pontot, vagy közli, hogy ez bizony nem a szólóban levő lap volt, tévedtél.

A játék során lehetőség van az „asztal alá nézni"! Képzeld el, hogy a lapok egy üvegasztalon vannak, s ha az asztal alá nézel, megleshed az összes lapot.

Ha ezt akarod tenni, úgy a gép kérdésére, hogy melyik lapot kívánod megnézni, a megfelelő sorszám helyett egy A betűt kell válaszolnod! Erre a gép felfordítja az összes lapot sorban, majd ugyanebben a sorrendben vissza is fordítja. Mindez természetesen rövid idő alatt történik, mindenesetre néhány lap helyét jól meg tudod jegyezni ez idő alatt. Épp ezért nagy árat is kell fizetned az asztal alá nézésért. A játék során az első esetben 300, a második esetben 4×300, a harmadik esetben 9×300 pontot von le tőled a gép a kíváncsiságért!

A játék lejátszásához, valamennyi lap levételéhez optimális esetben minimálisan 18 lépésre van szükség. Amennyiben ezen a lépésszámon már túl vagy, úgy a gép minden további lépéspárért 50 pontot von le tőled. Ezt se felejtsd el! Viszont, ha minden kártyát sikerül „levenned" 8 perc alatt, ezért plusz jutalompontokat kapsz a megspórolt időtől függően. A jutalom minimális összege 500 pont! Nyolc perc után a játék automatikusan leáll, kiírja a pontszámodat, s folytatja egy újabb feladattal, amelynek játékszabályát majd a képernyőről olvashatod el!

Jó játékot!"

Mielőtt a Memory játék programjának ismertetésére térnénk, közölnünk kell, hogy sajnos a program átírásánál nem jártunk teljes sikerrel. A HT1080Z, a Commodore-64 és a Sinclair Spectrum gépeken nem tudtuk elérni, hogy 35 lap férjen el a képernyőn, ugyanis a kártyalapokon el kell férnie a 8 karakter hosszúságú szövegeknek is. Így hát az előző játékszabály a többi gépen némileg módosul. Ezek a módosítások a listákban levő szabályból kiolvashatók. (Figyelem! HT1080Z gépen a játékhoz szükség van a tárbővítésre!)

Bízva abban, hogy ezt a — könyvünkben is kiemelkedően leghosszabb — programot néhányan azért beírják gépükbe, s kipróbálva beleszeretnek a véleményünk szerint egészen jól sikerült játékba, úgy gondoltuk, hogy főleg azokat, akik játszanak is vele, érdekli, hogyan is áll össze ez a kicsit bonyolult programocska. Így hát a következő oldalakon igyekszünk nem túlságosan bőven, de azért elegendő magyarázattal szolgálni. Sajnos azonban, aki soha nem látta és nem is játszotta játékunkat, bizonyos dolgokat nehezen ért majd meg magyarázatunkból. Ezért valóban érdemes előbb kipróbálni, s azután megérteni!

```
1    ##### MEMORY PROGRAM #####
10 DEFINT E-J,L:DEFSTR O-R
20 DIM PU(35),EU(35),R(5,7),E(5,7)
30
40    ##### ADATOK, KONSTANSOK INICIALIZALASA #####
50
60 MAXIDO=14000
70 CC=15360
80 DEF FNK(I,J)=CC+(I-1)*80+J-1
90 R$(0)="  SCHL  ":R$(1)="  SZKI  "
100 IDB=35:C1=.1:C2=50:C3=300:C4=500 :  ##### PONTOZASI KONSTANSOK.
FELHASZNALASUKAT LASD A KOV. SOROKBAN:   4020,11030,
13020.    ######
110 IN=127:IB=144:IK=129:RN=CHR$(IN)
120 RSP=SPACE$(77):RS0=SPACE$(10)
130
140    ##### KARTYAFELIRATOK ES ERTEKEI #####
150
```

```
160 DATA BOEINGa,100,Bo ingb,100,stringc,100,bugyutad,100,Morickae,100,Moriczkaf
,100
170 DATA BOEINGa,100,Bo ingb,100,stringc,100,bugyutad,100,Morickae,100,Moriczkaf
,100
180 DATA WERTYUg,150,TLEHRAPSi,150,M08XH,150,SCITELE,150,PROPERF,150
190 DATA WERTYUg,150,TLEHRAPSi,150,M08XH,150,SCITELE,150,PROPERF,150
200 DATA 3+6/3*21,250,3+6/3/2m,250,6-3/2*2n,250,16/(2*4)o,250,16/2*4p,250,7*37-2
59q,250
210 DATA 71,250,4m,250,3n,250,2o,250,32p,250,0q,250,M08Xr,300
220 '
230 ' ##### A JATEK KEZDETE #####
240 '
250 PRINT CHR$(17):PRINT:PRINT:PRINT:PRINT"                                    M E M O
 R Y":PRINT:PRINT
260 PRINT"Kersz jatekszabalyt? ";
270 IDO=IDO+1:Q=INKEY$:IF Q="" THEN 270 ELSE PRINT Q;:IF Q="n" OR Q="N" THEN PRI
NT:PRINT:PRINT"Egy picike turelmet kerek, ugyanis mo
st keverem a zsugat.":PRINT:PRINT:PRINT ELSE GOSUB 12000
280 RANDOMIZE IDO:GOSUB 8000
290 PRINT"Kezdhetem? (Ha igen, nyomd meg a betukoz billentyut!)";
300 Q=INKEY$
310 IDO=IDO+1:IF IDO>MAXIDO THEN 11000
320 Q=INKEY$:IF Q="" THEN 310
330 GOSUB 1000
340 GOSUB 5000:GOSUB 2000:GOTO 340
990 '
1000 ' ### SUBR. ELH. ###
1010 '
1020 PRINT CHR$(17):FOR I=2 TO 20 STEP 2:POKE FNK(I,78),IK
1030 FOR J=1 TO 67 STEP 11:POKE FNK(I,J),IB:NEXT J,I
1040 FOR SOR=2 TO 18 STEP 4:FOR OSZ=1 TO 67 STEP 11:GOSUB 10000:PRINT RN+RS((SOR
=6 OR SOR=14)+OSZ AND 1)+RN;:POKE FNK(SOR+1,OSZ+11),
IK:NEXT OSZ,SOR
1050 FOR OSZ=5 TO 71 STEP 11:SOR=0:GOSUB 10000:PRINT 1+INT(OSZ/11):NEXT OSZ
1060 FOR SOR=2 TO 18 STEP 4:OSZ=78:GOSUB 10000:PRINT MID$(STR$(1+INT(SOR/4)),2);
:NEXT SOR:RETURN
1990
2000 ' ### JO TIPP MEGVAL. ###
2010 '
2020 PR=PR+1
2030 SOR=23:OSZ=0:GOSUB 10000:PRINT"Ha megnezted, nyomd meg a betukoz billentyut
!                              ";
2040 Q=INKEY$
2050 IDO=IDO+1:IF IDO>MAXIDO THEN 11000
2060 Q=INKEY$:IF Q="" THEN 2050
2070 SOR=20:OSZ=0:GOSUB 10000:PRINT RSP:PRINT RSP:PRINT RSP:PRINT RSP;:GOSUB 100
00

2080 FOR K=1 TO 2:POKE FNK(II(K)*4-1,JJ(K)*11-10),IK:NEXT K
2090 IF R(II(1),JJ(1))=RM AND R(II(2),JJ(2))=RM THEN 2180
2100 IF MID$(R(II(1),JJ(1)),9)=MID$(R(II(2),JJ(2)),9) THEN 2130
2110 PRINT"Ez sajnos ket kulonbozo volt!

2120 FOR K=1 TO 2:II=II(K):JJ=JJ(K):GOSUB 9000:PRINT RS(II+JJ+1 AND 1);:NEXT K:R
ETURN
2130 PRINT"Ez a ket lap a tied!
    ":FOR K=1 TO 2
2140 POKE FNK(II(K)*4,JJ(K)*11-9),IK:POKE FNK(II(K)*4-2,JJ(K)*11-9),IK
2150 SOR=II(K)*4-2:OSZ=JJ(K)*11-10:GOSUB 10000:PRINT RS0;
2160 PONT=PONT+E(II(K),JJ(K))/2:E(II(K),JJ(K))=0:NEXT K
2170 IDB=IDB-2:IF IDB=0 THEN 13000 ELSE RETURN
2180 PRINT"Kitalaltad, ennek a lapnak nincs parja, ugyhogy ez a tied!"
2190 II=II(1):JJ=JJ(1):POKE FNK(II*4,JJ*11-9),IK:POKE FNK(II*4-2,JJ*11-9),IK
2200 SOR=II*4-2:OSZ=JJ*11-10:GOSUB 10000:PRINT RS0;
2210 PONT=PONT+E(II,JJ):E(II,JJ)=0
2220 IDB=IDB-1:IF IDB=0 THEN 13000 ELSE RETURN
2990 '
```

```
3000 ' ### EGY KARTYA FELFORD. ###
3010 '
3020 POKE FNK(II(K2)*4-1,JJ(K2)*11-10),IB
3030 II=II(K2):JJ=JJ(K2):GOSUB 9000
3040 PRINT LEFT$(R(II,JJ),8);:RETURN
3990 '
4000 ' ### ASZTAL-ALA NEZES ###
4010 '
4020 IVIL=IVIL+1:PONT=PONT-C3*IVIL*IVIL
4030 SOR=20:OSZ=0:GOSUB 10000:PRINT RSP:PPINT RSP:PRINT RSP:PRINT RSP;
4040 FOR II=1 TO 5:FOR JJ=1 TO 7
4050 IF E(II,JJ)<>0 THEN POKE FNK(II*4-1,JJ*11-10),IB:GOSUB 9000:PRINT LEFT$(R(I
I,JJ),8);
4060 NEXT JJ,II
4070 FOR II=1 TO 5:FOR JJ=1 TO 7
4080 IF E(II,JJ)<>0 THEN POKE FNK(II*4-1,JJ*11-10),IK:GOSUB 9000:PRINT F3(II+JJ+
1 AND 1);
4090 NEXT JJ,II:GOTO 340
4990 '
5000 ' ### TIPP BE ###
5010 '
5020 SOR=21:OSZ=0:GOSUB 10000:PRINT"Az elso  lap hanyadik  sorban van? ";
5030 KAP=1:GOTO 6000
5040 II(1)=IQ
5050 SOR=20:OSZ=0:GOSUB 10000:PRINT RSP;:SOR=23:GOSUB 10000:PRINT RSP;:SOR=21
5060 OSZ=40:GOSUB 10000:PRINT"Az elso  lap hanyadik  oszlopban van? ";
5070 KAP=2:GOTO 6000
5080 JJ(1)=IQ
5090 K2=1:GOTO 7000
5100 GOSUB 3000
5110 OSZ=0:SOR=22:GOSUB 10000:PRINT"A masodik lap hanyadik sorban van? ";
5120 KAP=3:GOTO 6000
5130 II(2)=IQ
5140 OSZ=40:GOSUB 10000:PRINT"A masodik lap hanyadik oszlopban van? ";
5150 KAP=4:GOTO 6000
5160 JJ(2)=IQ
5170 K2=2:GOTO 7000
5180 GOSUB 3000
5190 IF II(1)<>II(2) OR JJ(1)<>JJ(2) OR R(II(1),JJ(1))=RM THEN RETURN
5200 SOR=23:OSZ=0:GOSUB 10000:PPINT"Tevedtel, ez nem a paratlan lap
                                        "  :PP=PP+1
5210 OKE FNK(II(1)*4-1,JJ(1)*11-10),IK
5220 GOSUB 2120:SOP=21:OSZ=0:GOSUB 10000:PRINT RSP;
5230 SOR=22:OSZ=0:GOSUB 10000:PPINT RSP;:GOTO 5020
5990 '
6000 ' ### TIPP ELL. ###
6010 '
6020 Q=INKEY$
6030 IDO=IDO+1:IF IDO>MAXIDO THEN 11000
6040 Q=INKEY$:IF Q="" THEN 6030
6050 IF Q="a" OR Q="A" THEN 4000
6060 IF Q>"0" AND ASC(Q)<56-(KAP AND 1)*2 THEN PRINT Q;:IQ=VAL(Q):ON KAP GOTO 50
40,5080,5130,5160
6070 ON KAP GOTO 5020,5060,5110,5140
6990 '
7000 ' ### NEMLETEZO LAP KISZURESE ###
7010 '
7020 IF E(II(K2),JJ(K2))>0 THEN ON K2 GOTO 5100,5180
7030 S3=SOR:SOR=23:OSZ=0:GOSUB 10000:PRINT"Ezen a helyen mar nincs lap!
                                        ";
7040 SOR=S3:OSZ=0:GOSUB 10000:PRINT RSP;
7050 ON K2 GOTO 5020,5110
7990 '
8000 ' ### KEVERES ###
8010 '
8020 FOR I=1 TO 35:READ RM,EV(I)
8030 R1=LEFT$(RM,LEN(RM)-1):R2=RIGHT$(RM,1)
```

```
8040 L=8-LEN(R1):RM=SPACE$(INT((L+1)/2))+R1+SPACE$(INT(L/2))+R2:RV(I)=RM:NEXT I
8050 FOR I=1 TO 10
8060 L1=1+INT(RND*35):L2=1+INT(RND*35):SWAP RV(L1),RV(L2):SWAP EV(L1),EV(L2):NEX
T I
8070 PK=1+INT(RND*35):S=RND:P=11+(S<2/3)*2+(S<1/3)*6:FOR I=1 TO 5:FOR J=1 TO 7
8080 PK=PK+P:IF PK>35 THEN PK=PK-35
8090 R(I,J)=RV(PK):E(I,J)=EV(PK):NEXT J,I
8100 FOR I=1 TO 12:I1=1+INT(5*RND):J1=1+INT(7*RND):I2=1+INT(5*RND):J2=1+INT(7*RN
D)
8110 SWAP R(I1,J1),R(I2,J2):SWAP E(I1,J1),E(I2,J2):NEXT I
8120 RETURN
8990 '
9000   ### SUBR. II. SOR JJ. KARTYAJA ###
9010 '
9020 SOR=II*4-2:OSZ=JJ*11-9:GOSUB 10000:RETURN
9980 '
9990 REM  ############################
10000 REM #### SUBR. KURZOR POZ. #####
10010 REM #### SOR=Sorszam (0-24) ####
10020 REM #### OSZ=Oszlopszam (0-79) #
10030 REM #### Hasznalt valtozok: ####
10040 REM #### S1, S2, O1, O2, AAA$. #
10050 REM ############################
10060 SOR=INT(ABS(SOR)):OSZ=INT(ABS(OSZ))
10070 IF SOR<0 OR SOR>24 OR OSZ<0 OR OSZ>79 THEN RETURN
10080 S1=INT(SOR/10)+48:S2=SOR MOD 10+48:O1=INT(OSZ/10)+48:O2=OSZ MOD 10+48
10090 AAA$="["+CHR$(S1)+CHR$(S2)+CHR$(O1)+CHR$(O2)+"f"
10100 PRINT CHR$(27)
10110 PRINT CHR$(27);AAA$;
10120 PRINT CHR$(27)
10130 RETURN
10990 '
11000 ' ### VEGE ###
11010 '
11020 PRINT CHR$(17):PRINT:PRINT:PRINT"           SAJNOS A JATEKIDOD LEJART!"
11030 PONT=INT(PONT-(PR-18)*C2)
11040 PRINT:PRINT:PRINT"A jatekban elert pontszamod:";PONT
11050 END
11990 '
12000 ' ### JATEKSZABALYOK ###
12010 '
12020 PRINT CHR$(17):PRINT:PRINT"                    M E M O R Y   J A T E K"
12030 PRINT:PRINT"A kepernyon 35 db. kartyat fogsz latni, persze leforditva."
12040 PRINT"A kartyak hatoldalan 'SZKI', ill. 'Sci-L' feliratok vannak."
12050 PRINT"Egymas utan felforditহatsz ket kartyat, s ha egyformak, vagy pedig"
12060 PRINT"az egyiken egy kifejezes, a masikon pedig ennek szamerteke lathato,"
12070 PRINT"akkor a ket kartyat megkapod. Persze van egy kartya, amelyiknek ninc
s
12080 PRINT"parja. Ha rajottel, hogy melyik ez, akkor elsonek is es masodiknak i
s"
12090 PRINT"ezt felforditva megkaphatod."
12100 PRINT"Egy kartyapar 100 (ertelmes szo), 150 (ertelmetlen), vagy 250 (kifej
ezes)"
12110 PRINT"pontot er, az egyedul allo kartya 300-at. A 35 kartyat 18 probalkoza
sbol"
12120 PRINT"el lehet venni, az efolotti probalkozasokert a gep 50-50 pontot von
le."
12130 PRINT"Az egesz jatekra (ennek a szovegnek az elolvasaval egyutt) kb. 8 per
ced van,"
12140 PRINT"ha ennel hamarabb leveszed az osszes kartyat, akkor tobb, mint 500 "
12150 PRINT"(az idotol is fuggo) jutalompontot kapsz."
12160 PRINT"A kartyak egy uveglapon vannak, s ha a tipp helyett egy 'A' zetut ut
sz be,"
12170 PRINT"akkor benezhetsz az asztal ala, rovid ideig megnezheted az osszes ka
rtya "
12180 PRINT"szinet. Ezert azonban nagy arat kell fizetned, az elso ilyenert 300,
"
```

```
12210 RETURN
12990 '
13000 ' ### HA MIND LEVETTE ###
13010 '
13020 PON=INT(C4+(MAXIDO-IDO)*C1)
13030 PRINT CHR$(17):PRINT:PRINT:PRINT"          GRATULALOK, MINDEN LAPOT SIKERUL
T LEVENNED.":PRINT:PRINT"        Ezert";PON;"jutalo
mpontot kaptal.":PONT=PONT+PON:GOTO 11030
```

```
1 '########## M E M O R Y ############
10 CLEAR 1000:DEFSTR Q-R:DEFINT H-N,E,X,Y
20 DIM RV(24),EV(24),R(4,6),E(4,6)
25 '
30 '####### KONSTANSOK INICIALIZALASA #########
40 MI=7800
50 RS(1)="  Sci-L ":RS(0)="  Szki  "
60 D8=24:C1=.1:C2=50:C3=300:C4=500
70 DATA 168,140,172,170,32,170,130,131,131
80 FOR K=1 TO 3:READ A,B,C:FOR I=1 TO 6:Q(K)=Q(K)+CHR¤(A):FOR J=1 TO 8:Q(K)=Q(K)
+CHR¤(B):NEXT J:Q(K)=Q(K)+CHR¤(C):NEXT I,K
100 QS=CHR¤(202):RS=CHR¤(255)
110 '
120 '###### KARTYAFELIRATOK éS éRTéKEIK ######
130 DATA stringA,100,STRINGC,100,MORICKAE,100,MORICZKAF,100
140 DATA stringA,100,STRINGC,100,MORICKAE,100,MORICZKAF,100
150 DATA wertyug,150,TLEHRAPSI,150,M08XH,150,M08XJ,150
160 DATA wertyug,150,TLEHRAPSI,150,M08XH,150,M08XJ,150
170 DATA 3+6/3*2L,250,3+6/3/2M,250,16/(2*4)O,250,16/2*4P,250
180 DATA 7L,250,4M,250,2O,250,32P,250
190 '
200 '########### A JATéK KEZDETE ############
210 CLS:PRINT:PRINT:PRINT"                    M E M O R Y":PRINT:PRINT
220 PRINT "KéRSZ JATéKSZBALYT? "::Q=INKEY¤
230 IDO=IDO+1:Q=INKEY¤:IF Q="" THEN 230
240 PRINT Q:IF Q<>"N" THEN GOSUB 13000:GOTO 260
250 PRINT:PRINT "EGY PICIKE TüRELMET KéREK, UGYANIS MOST KEVEREM A ZSUGAT.":PRIN
T
260 GOSUB 8000
270 PRINT "KEZDHETEM? (HA IGEN, NYOMD MEG A BETüKöZ BILLENTYüT!)";:Q=INKEY¤
280 IDO=IDO+1:Q=INKEY¤:IF Q="" THEN 280
290 GOSUB 1000
300 GOSUB 5000:GOSUB 2000:GOTO 300
990 '
1000 '##### KARTYAK ELHELYEZéSE #####
1010 CLS:PRINT:FOR I=1 TO 4:FOR J=1 TO 3:PRINT Q(J):NEXT J,I
1020 FOR Y=1 TO 51 STEP 10:FOR X=2 TO 11 STEP 3
1030 GOSUB 10000:PRINT éX*64+Y,RS(P);:NEXT X
1040 PRINT éY+3,1+INT(Y/10);:NEXT Y
1050 FOR X=2 TO 11 STEP 3:PRINT éX*64+61,1+INT(X/3);:NEXT X
1060 RETURN
1990 '
2000 '###### JO TIPP MEGVALOSITASA #######
2010 PR=PR+1:PRINT é960,"HA MEGNéZTED, NYOMD MEG A BETüKöZT!
     ";:Q=INKEY¤
2020 IDO=IDO+1:IF IDO>MI THEN 12000
2030 Q=INKEY¤:IF Q="" THEN 2020
2040 PRINT é832,RS;:PRINT é896,RS
2050 IF MID¤(R(II(1),JJ(1)),9)=MID¤(R¤(II(2),JJ(2)),9) THEN 2080
2060 PRINT é960,"EZ SAJNOS KéT KüLöNBöZö VOLT!
:
```

```
2070 FOR K=1 TO 2:GOSUB 9000:PRINT éXY,RS(P);:NEXT K:RETURN
2080 PRINT é960,"EZ A KéT LAP A TIED!
;
2090 FOR K=1 TO 2:GOSUB 9000:FOR I=-1 TO 1:PRINT éXY-1+I*64,QS;:NEXT I
2100 PO=PO+E(II(K),JJ(K))/2:E(II(K),JJ(K))=0
2110 NEXT K:DB=DB-2:IF DB=0 THEN 14000
2120 RETURN
2990 '
3000 '###### EGY KARTYA FELFORDITASA #######
3010 GOSUB 9000:PRINT éXY,LEFT¤(R(II(K),JJ(K)),8);:RETURN
3990 '
4000 '####### ASZTAL ALA NéZéS ########
4010 VIL=VIL+1:PO=PO-C3*VIL*VIL
4020 PRINT é832,RS;:PRINT RS:PRINTRS;
4030 FOR II=1 TO 4:FOR JJ=1 TO 6
4040 IF E(II,JJ)<>0 THEN GOSUB 9500:PRINT éXY,LEFT¤(R(II,JJ),8);
4050 NEXT JJ,II
4060 FOR II=1 TO 2000:NEXT II
4070 FOR II=1 TO 4:FOR JJ=1 TO 6
4080 IF E(II,JJ)<>0 THEN GOSUB 9500:PRINT éXY,RS(P);
4090 NEXT JJ,II:GOTO 300
4990 '
5000 '########## TIPP BEKéRDEZéSE ###########
5010 PRINT é832,"AZ ELSŐ LAP SORA?   ";
5020 KAP=2:GOSUB 6000
5030 II(1)=IQ
5040 PRINT é896,RS:PRINTRS;
5050 PRINT é864,"AZ ELSŐ LAP OSZLOPA?   ";
5060 KAP=0:GOSUB 6000
5070 JJ(1)=IQ:K=1:GOTO 7000
5080 GOSUB 3000
5090 PRINT é896,"A MASODIK LAP SORA? ";
5100 KAP=2:GOSUB 6000
5110 II(2)=IQ
5120 PRINT é928,"A MASODIK LAP OSZLOPA? ";
5130 KAP=0:GOSUB 6000
5140 JJ(2)=IQ:K=2:GOTO 7000
5150 GOSUB 3000
5160 IF II(1)<>II(2) OR JJ(1)<>JJ(2) THEN RETURN
5170 PRINT é960,"KéT KÜLONBÖZŐ LAPOT KELL MEGADNOD!
;
5180 PRINT é896,RS;:GOTO 5090
5990 '
6000 '####### TIPP ELLENŐRZéSE ########
6010 Q=INKEY¤
6020 IDO=IDO+1:IF IDO>MI THEN 12000
6030 Q=INKEY¤:IF Q="" THEN 6020
6040 IF Q="A" THEN 4000
6050 IF Q<"1" OR ASC(Q)>54-KAP THEN 6020
6060 PRINT Q;:IQ=VAL(Q):RETURN
6990 '
7000 '##### LEVETT LAP KISZÜRéSE #####
7010 IF E(II(K),JJ(K))>0 THEN ON K GOTO 5080,5150
7020 PRINT é960,"EZEN A HELYEN MAR NINCS LAP!
;
7030 PRINT é768+K*64,RS;:ON K GOTO 5010,5090
7990 '
8000 '############## KEVERéS ###############
8010 FOR I=1 TO DB:READ RM,EV(I)
8020 Q1=LEFT¤(RM,LEN(RM)-1):Q2=RIGHT¤(RM,1)
8030 L=8-LEN(Q1)
8040 RV(I)=LEFT¤("     ",INT((L+1)/2))+Q1+LEFT¤("     ",INT(L/2))+Q2:NEXT I
8050 FOR I=1 TO 10:L1=RND(DB):L2=RND(DB)
8060 RR=RV(L1):RV(L1)=RV(L2):RV(L2)=RR:EE=EV(L1):EV(L1)=EV(L2):EV(L2)=EE
8070 NEXT I:PK=RND(DB)
8080 P=RND(3)*6-3+2*RND(2)
8090 FOR I=1 TO 4:FOR J=1 TO 6:PK=PK+P:IF PK>DB THEN PK=PK-DB
```

```
8100 R(I,J)=RV(PK):E(I,J)=EV(PK):NEXT J,I
8110 FOR I=1 TO 10:I1=RND(4):I2=RND(4):J1=RND(6):J2=RND(6)
8120 RR=R(I1,J1):R(I1,J1)=R(I2,J2):R(I2,J2)=RR
8130 EE=E(I1,J1):E(I1,J1)=E(I2,J2):E(I2,J2)=EE
8140 NEXT I:RETURN
8990 '
9000 '##### II(K). SOR JJ(K). KARTYAJA #####
9010 II=II(K):JJ=JJ(K)
9500 X=3*II-1:Y=10*JJ-9:XY=X*64+Y
10000 P=((X AND 1)+(INT(Y/10) AND 1)) AND 1:RETURN
11990 '
12000 '############ VÉGE ##############
12010 CLS:PRINT:PRINT:PRINT "                      SAJNOS A JATÉKIDŐD LEJART!":PRINT
12020 PO=INT(PO+(PR-12)*C2*(PR)12)):PD=-PO*(PO)0)
12030 PRINT:PRINT:PRINT "A JATEKBAN ELÉRT PONTSZAMOD:";PO;"PONT.":PRINT:PRINT
12040 END
12990 '
13000 '####### JATÉKSZABALY ########
13010 CLS:PRINT "           M E M O R Y  -  J A T É K S Z A B A L Y":PRINT
13020 PRINT "A KÉPERNYŐN 24 KARTYAT FOGSZ LATNI, PERSZE LEFORDITVA."
13030 PRINT "A KARTYAK 12 PART ALKOTNAK, AZ EGY PARBA TARTOZOK EGYFORMAK,"
13040 PRINT "VAGY AZ EGYIKEN LEVŐ KIFEJEZÉS ÉRTÉKE EGYENLŐ A MASIKON LEVŐ"
13050 PRINT "SZAMMAL. NEKED EZEKET A PAROKAT KELL MEGTALALNOD."
13060 PRINT "MINDIG KÉT KARTYAT FORDITHATSZ FEL, S HA KÜLÖNBÖZŐK, AKKOR VISZ-SZA
FORDULNAK. HA VISZONT EGYFORMAK, AKKOR LEVEHETED ŐKET, S"
13070 PRINT "KAPSZ ÉRTÜK 100-250 PONTOT. AMENNYIVEL TÖBBET PROBALGATSZ"
13080 PRINT "12-NÉL, ANNYISZOR 50 PONTOT LEVONOK TŐLED. VISZONT HA MINDEN"
13090 PRINT "KARTYAT LEVESZEL IDŐN (6 PERC) BELÜL, AKKOR TÖBB, MINT 500"
13100 PRINT "JUTALOMPONTOT KAPSZ. A KARTYAK EGY ÜVEGASZTALON VANNAK, S NEKED"
13110 PRINT "LEHETŐSÉGED VAN AZ ASZTAL ALA NÉZNI, CSAK EGY A BETŰT KELL BE-"
13120 PRINT "ÜTNÖD. EZÉRT AZONBAN NAGY ARAT FIZETSZ, AZ ELSŐ ALKALOMMAL 1,"
13130 PRINT "MASODSZOR 4, HARMADSZOR 9, STB. -SZER 300 PONTOT VONOK LE."
13140 RETURN
13990 '
14000 '####### HA MIND LEVETTE ########
14010 P0=INT(C4+(MI-IDO)*C1):PO=PO+P0
14020 CLS:PRINT:PRINT:PRINT"         GRATULALOK, MINDENT LE TUDTAL VENNI!"
14030 PRINT:PRINT:PRINT "EZÉRT";P0;"JUTALOMPONTOT KAPTAL.":GOTO 12020
```

Megjegyzés
Azoknál a HT1080Z gépre készült programoknál, amelyekben ékezetes betűk is szerepelnek, kénytelenek voltunk az ún. svéd karakterkészletet használni a nyomtatás során. Ezekben a programokban tehát dollárjel helyett ¤, és mint ahogy az lenni szokott, a @ jel helyett É szerepel.

```
1 REM ########## M E M O R Y ###########
10 POKE 251,24:POKE 252,96:SYS 251
15 DEF FN PAR(X)=(X=6 OR X=14)+(Y=1 OR Y=21) AND 1
20 DIM RV$(20),EV(20),R$(5,4),E(5,4)
29 REM
30 REM ### KONSTANSOK INICIALIZALASA ###
40 MAXIDO=11000
50 RS$(0)="  SCI-L ":RS$(1)="  SZKI  "
60 DB=20:C1=.1:C2=50:C3=300:C4=500
70 R1$="╭────────╮╭────────╮╭────────╮╭────────╮"
80 R2$="│        ││        ││        ││        │"
90 R3$="╰────────╯╰────────╯╰────────╯╰────────╯"
100 PS0$="                  ":RSP$="                                        "
110 REM
120 REM ## KARTYAFELIRATOK ES ERTEKEIK #
130 DATA STRINGC,100,MORICKAE,100,MORICZKAF,100
```

```
140 DATA STRINGC,100,MORICKAE,100,MORICZKAF,100
150 DATA WERTYUG,150,TLEHRAPSI,150,M08XH,150
160 DATA WERTYUG,150,TLEHRAPSI,150,M08XH,150
170 DATA 3+6/3*2L,250,3+6/3/2M,250,16/(2*4)O,250,16/2*4P,250
180 DATA 7L,250,4M,250,2O,250,32P,250
190 REM
200 REM ####### A JATEK KEZDETE ########
210 PRINT"⎕":PRINT:PRINT:PRINT"                M E M O R Y":PRINT:PRINT
220 PRINT "KERSZ JATEKSZABALYT? ";
225 POKE 198,0
230 IDO=IDO+1:GET Q$:IF Q$="" THEN 230
240 PRINT Q$:IF Q$<>"N" THEN GOSUB 13000:GOTO 260
250 PRINT:PRINT"EGY PICIKE TURELMET KEREK, UGYANIS MOST KEVEREM A ZSUGAT.":PRINT
260 GOSUB 8000
270 PRINT "KEZDHETEM? (HA IGEN, NYOMD MEG A        BETUKOZ BILLENTYUT!)";
275 POKE 198,0
280 IDO=IDO+1:GET Q$:IF Q$="" THEN 280
290 GOSUB 1000
300 GOSUB 5000:GOSUB 2000:GOTO 300
999 REM
1000 REM ##### KARTYAK ELHELYEZESE #####
1010 PRINT "⎕":FOR I=0 TO 4:PRINT R1$;R2$;R3$:NEXT I
1020 FOR Y=1 TO 31 STEP 10:FOR X=2 TO 18 STEP 4:GOSUB 9999
1030 PRINT RS$(FN PAR(X));:NEXT X,Y
1040 RETURN
1999 REM
2000 REM #### JO TIPP MEGVALOSITASA ####
2010 PR=PR+1:X=24:Y=0:GOSUB 9999:PRINT "HA MEGNEZTED, NYOMD MEG A BETUKOZT!   ";
2015 POKE 198,0
2020 IDO=IDO+1:IF IDO>MAXIDO THEN 12000
2030 GET Q$:IF Q$="" THEN 2020
2040 Y=0:FOR X=20 TO 24:GOSUB 9999:PRINT RSP$;:NEXT X
2050 X=20:GOSUB 9999
2060 IF MID$(R$(II(1),JJ(1)),9)=MID$(R$(II(2),JJ(2)),9) THEN 2090
2070 PRINT "EZ SAJNOS KET KULONBOZO VOLT!          "
2080 FOR K=1 TO 2:GOSUB 9000:PRINT RS$(FN PAR(X));:NEXT K:RETURN
2090 PRINT "EZ A KET LAP H TIED!                    "
2100 FOR K=1 TO 2:GOSUB 9000:Y=Y-1:GOSUB 9999:PRINT PS0$;:X=X-1:GOSUB 9999
2105 PRINT PS0$;
2110 X=X+2:GOSUB 9999:PRINT PS0$;:PO=PO+E(II(K),JJ(K))/2:E(II(K),JJ(K))=0
2120 NEXT K:DB=DB-2:IF DB=0 THEN 14000
2130 RETURN
2990 REM
3000 REM ### EGY KARTYA FELFORDITASA ###
3010 GOSUB 9000:PRINT LEFT$(R$(II(K),JJ(K)),8);:RETURN
3990 REM
4000 REM ####### ASZTAL ALA NEZES ######
4010 VIL=VIL+1:PO=PO-C3*VIL*VIL
4020 Y=0:FOR X=20 TO 24:GOSUB 9999:PRINT RSP$;:NEXT X
4030 FOR II=1 TO 5:FOR JJ=1 TO 4
4040 IF E(II,JJ)<>0 THEN GOSUB 9500:PRINT LEFT$(R$(II,JJ),8);
4042 NEXT JJ,II
4045 FOR I=1 TO 5000:NEXT I
4050 FOR II=1 TO 5:FOR JJ=1 TO 4
4060 IF E(II,JJ)<>0 THEN GOSUB 9500:PRINT RS$(FN PAR(X));
4070 NEXT JJ,II:GOTO 300
4990 REM
5000 REM ####### TIPP BEKERDEZESE ######
5010 X=21:Y=0:GOSUB 9999:PRINT " AZ 1. LAP SORA? ";
5020 KAP=1:GOSUB 6000
5030 II(1)=IO
5040 X=20:GOSUB 9999:PRINT RSP$;:X=23:GOSUB 9999:PRINT RSP$;:PRINT RSP$;:X=21
5050 Y=19:GOSUB 9999:PRINT "AZ 1. LAP OSZLOPA? ";
```

```
5060 KAP=0:GOSUB 6000
5070 JJ(1)=IQ:K=1:GOTO 7000
5080 GOSUB 3000
5090 X=22:Y=0:GOSUB 9999:PRINT " A,2. LAP SORA?  ";
5100 KAP=1:GOSUB 6000
5110 II(2)=IQ
5120 Y=19:GOSUB 9999:PRINT "A 2. LAP OSZLOPA?  ";
5130 KAP=0:GOSUB 6000
5140 JJ(2)=IQ:K=2:GOTO 7000
5150 GOSUB 3000
5160 IF II(1)<>II(2) OR JJ(1)<>JJ(2) THEN RETURN
5170 X=24:Y=0:GOSUB 9999:PRINT "KET KULONBOZO LAPOT KELL MEGADNOD!";
5180 X=22:Y=0:GOSUB 9999:PRINT RSP$;:GOTO 5090
5990 REM
6000 REM ####### TIPP ELLENORZESE ######
6005 POKE 198,0
6010 IDO=IDO+1:IF IDO>MAXIDO THEN 12000
6020 GET Q$:IF Q$="" THEN 6010
6030 IF Q$="A" THEN 4000
6040 IF Q$<"1" OR ASC(Q$)>52+KAP THEN 6010
6050 PRINT Q$;:IQ=VAL(Q$):RETURN
6990 REM
7000 REM ### NEMLETEZO LAP KISZURESE ###
7010 IF E(II(K),JJ(K))>0 THEN ON K GOTO 5080,5150
7020 X1=X:X=24:Y=0:GOSUB 9999:PRINT "ITT MAR NINCS LAP!!!
7030 X=X1:GOSUB 9999:PRINT RSP$;:ON K GOTO 5010,5090
7990 REM
8000 REM ########### KEVERES ###########
8010 RW=RND(0):FOR I=1 TO 20:READ RM$,EV(I)
8020 P1$=LEFT$(RM$,LEN(RM$)-1):P2$=RIGHT$(RM$,1)
8030 L=8-LEN(P1$)
8040 RV$(I)=LEFT$("     ",INT((L+1)/2))+P1$+LEFT$("     ",INT(L/2))+P2$:NEXT I
8050 FOR I=1 TO 10:L1=1+INT(RND(1)*20):L2=1+INT(RND(1)*20)
8060 RR$=RV$(L1):RV$(L1)=RV$(L2):RV$(L2)=RR$:EE=EV(L1):EV(L1)=EV(L2):EV(L2)=EE
8070 NEXT I:PK=1+INT(RND(1)*20)
8075 P=3+2*INT(RND(1)*8):IF P=5*INT(P/5) THEN 8075
8080 FOR I=1 TO 5:FOR J=1 TO 4:PK=PK+P:IF PK>20 THEN PK=PK-20
8090 R$(I,J)=RV$(PK):E(I,J)=EV(PK):NEXT J,I
8100 FOR I=1 TO 8:I1=1+INT(RND(1)*5):J1=1+INT(RND(1)*4)
8110 I2=1+INT(RND(1)*5):J2=1+INT(RND(1)*4)
8120 RR$=R$(I1,J1):R$(I1,J1)=R$(I2,J2):R$(I2,J2)=RR$
8130 EE=E(I1,J1):E(I1,J1)=E(I2,J2):E(I2,J2)=EE:NEXT I:RETURN
8990 REM
9000 REM ## II(K). SOR JJ(K). KARTYAJA #
9010 II=II(K):JJ=JJ(K)
9500 X=II*4-2:Y=JJ*10-9
9998 REM
9999 REM ##### KURZOR MOZGATO RUTIN ####
10000 IF X<0 OR X>24 OR Y<0 OR Y>39 THEN RETURN
10010 POKE 781,X:POKE 782,Y:SYS 65520:RETURN
11990 REM
12000 REM ########### VEGE #############
12010 PRINT "J":PRINT:PRINT:PRINT "     SAJNOS A JATEKIDOD LEJART!"
12020 PO=INT(PO+(PR-10)*C2*(PR>10)):PO=-PO*(PO>0)
12030 PRINT:PRINT:PRINT"A JATEKBAN ELERT PONTSZAMOD:";PO;"PONT."
12040 END
12990 REM
13000 REM ######### JATEKSZABALY #######
13010 PRINT "J":X=1:Y=14:GOSUB 9999:PRINT"M E M O R Y"
13020 PRINT "A KEPERNYON 20 KARTYAT FOGSZ LATNI,"
13030 PRINT "PERSZE LEFORDITVA. A KARTYAK 10 PART"
13040 PRINT "ALKOTNAK, AZ EGY PARBA TARTOZOK EGYFOR-"
13050 PRINT "MAK, VAGY AZ EGYIKEN LEVO KIFEJEZES ERT-KE EGYENLO A MASIKON LEVO";
13060 PRINT" SZAMMAL.":PRINT "NEKED EZEKET A PAROKAT KELL MEGTALALNOD.";
13070 PRINT "MINDIG KET KARTYAT FORDITHATSZ FEL, S HA KULONBOZOK, AKKOR";
13080 PRINT " VISSZAFORDULNAK.":PRINT"HA EGYFORMAK, LEVEHETED OKET, S KAPSZ"
13090 PRINT"ERTUK 100-250 PONTOT. AMENNYIVEL TOBBET"
```

```
13100 PRINT"PROBALGATSZ 10-NEL, ANNYISZOR 50 PONTOT"
13110 PRINT "LEVONOK TOLED. VISZONT HA MIND LEVESZED"
13120 PRINT "IDON (4 PERC) BELUL, AKKOR TOBB, MINT"
13130 PRINT "500 JUTALOMPONTOT KAPSZ."
13140 PRINT "A KARTYAK EGY UVEGASZTALON VANNAK, S"
13150 PRINT "NEKED LEHETOSEGED VAN AZ ASZTAL ALA NEZ-";
13160 PRINT "NI, CSAK EGY A BETUT KELL BEUTNOD."
13170 PRINT "EZERT AZONBAN NAGY ARAT FIZETSZ, AZ ELSOALKALOMMAL 300, MASODSZOR";
13180 PRINT " 4*300, HARMAD-";:PRINT "SZOR 9*300, STB. BUNTETOPONTOT VONOK LE."
13200 RETURN
13990 REM
14000 REM ####### HA MIND LEVETTE ######
14010 P0=INT(C4+(MAXIDO-IDO)*C1):PO=PO+P0
14020 PRINT"J":PRINT:PRINT:PRINT:PRINT "GRATULALOK, MINDENT LE TUDTAL VENNI!"
14030 PRINT:PRINT:PRINT "EZERT",P0;" JUTALOMPONTOT KAPTAL.":GOTO 12020
```

```
  1 REM #### M E M O R Y ####
  5 DEF FN Z(X)=X-2*INT (X/2)
 10 DEF FN P(X,Y)=1+FN Z(FN Z(X)+(Y=11))
 15 REM ### GRAFIKUS KARAKTEREK   DEFINIALASÂ ###
 20 DATA 24,24,0,0,0,24,24,24,0,0,0,24,12,48,0,0,0,24,15,240,3,
192,255,24,3,192,15,240,255,24,0,0,12,48,0,24,0,0,24,24,0,24,0,0
,24,24,0,24
 30 FOR I=0 TO 7: READ A,L,Q,P,G,F: POKE USR "A"+I,A: POKE USR
"L"+I,L: POKE USR "Q"+I,Q: POKE USR "P"+I,P: POKE USR "G"+I,G: P
OKE USR "F"+I,F: NEXT I
 40 DIM V$(18,9): DIM V(18): DIM R$(6,3,9): DIM E(6,3): DIM S$(
2,8): DIM I(2): DIM J(2)
 50 REM #### KONSTANSOK INICI- ALIZALASA ####
 60 LET MI=6000: LET ID=0: LET PR=0: LET VI=0: LET PO=0
 70 LET S$(2)="  Sci-L ": LET S$(1)="  Szki   "
 80 LET DB=18: LET C1=.1: LET C2=50: LET C3=300: LET C4=500
 90 LET W$=                         "
 |          ||          ||          |
"
100 LET T$="              ": LET B$="
"
110 REM ### KARTYAFELIRATOK ###
120 DATA "STRINGC",100,"MORICKAE",100,"MORICZKAF",100
130 DATA "STRINGC",100,"MORICKAE",100,"MORICZKAF",100
140 DATA "TLEHRAPSI",150,"M08XH",150
150 DATA "TLÈHRAPSI",150,"M08XH",150
160 DATA "3+6/3*2L",250,"3+6/3/2M",250,"16/(2*4)O",250,"16/2*4P
",250
170  DATA "7L",250,"4M",250,"20",250,"32P",250
180 REM ### A JATEK KEZDETE ###
190 CLS : PRINT : PRINT "            M E M O R Y": PRINT : PRINT
200 PRINT "KERSZ JATEKSZABALYT? ";: LET Q$=INKEY$
210 LET ID=ID+1: LET Q$=INKEY$: IF Q$="" THEN  GO TO 210
220 PRINT Q$: IF Q$<>"n" AND Q$<>"N" THEN  GO SUB 7500: GO TO 2
40
230 PRINT : PRINT "EGY PICIKE TURELMET KEREK, UGYA-NIS MOST KEV
EREM A ZSUGAT.": PRINT
240 GO SUB 8000
250 PRINT "KEZDHETEM? (HA IGEN, NYOMD MEG ABETUKOZ BILLENTYUT!)
";: LET Q$=INKEY$
260 LET ID=ID+1: LET Q$=INKEY$: IF Q$="" THEN  GO TO 260
270 GO SUB 1000
280 GO SUB 5000: GO SUB 2000: GO TO 280
1000 REM ### KARTYAK ELHELYEZESE      ###
1010 CLS : PRINT : FOR I=0 TO 5: PRINT W$: NEXT I
1020 FOR Y=1 TO 21 STEP 10: FOR X=2 TO 17 STEP 3
1030 PRINT AT X,Y;S$(FN P(X,Y));: NEXT X: NEXT Y
1040 FOR Y=0 TO 2: PRINT AT 0,Y*10+5;Y+1;: NEXT Y
```

```
1050 FOR X=0 TO 5: PRINT AT X*3+2,30;X+1;: NEXT X
1060 RETURN
2000 REM ### JO TIPP MEGVAL. ###
2010 LET PR=PR+1: PRINT AT 21,0;"NYOMD MEG A BETUKOZT!          ";
: LET Q$=INKEY$
2020 LET ID=ID+1: IF ID>MI THEN  GO TO 9800
2030 LET Q$=INKEY$: IF Q$="" THEN  GO TO 2020
2040 FOR X=19 TO 21: PRINT AT X,0;B$;: NEXT X
2050 IF R$(I(1),J(1),9)=R$(I(2),J(2),9) THEN  GO TO 2080
2060 PRINT AT 21,0;"EZ SAJNOS KET KULONBOZO VOLT! ";
2070 FOR K=1 TO 2: GO SUB 9000: PRINT AT X,Y;S$(FN P(X,Y));: NEX
T K: RETURN
2080 PRINT AT 21,0;"EZ A KET LAP A TIED!          ";
2090 FOR K=1 TO 2: GO SUB 9000: PRINT AT X-1,Y-1;T$;: PRINT AT X
,Y-1;T$;: PRINT AT X+1,Y-1;T$;
2100 LET PO=PO+E(I(K),J(K))/2: LET E(I(K),J(K))=0: NEXT K: LET D
B=DB-2: IF DB=0 THEN  GO TO 9600
2110 RETURN
3000 REM ### EGY KARTYA FELFORD.     ###
3010 GO SUB 9000: PRINT AT X,Y;R$(I(K),J(K),1 TO 8);: RETURN
4000 REM ### ASZTAL ALA NEZES ##
4010 LET VI=VI+1: LET PO=PO-C3*VI*VI
4020 FOR X=19 TO 21: PRINT AT X,0;B$;: NEXT X
4030 FOR I=1 TO 6: FOR J=1 TO 3
4040 IF E(I,J)<>0 THEN  GO SUB 9500: PRINT AT X,Y;R$(I,J,1 TO 8)
;
4050 NEXT J: NEXT I
4055 FOR H=1 TO 1000: NEXT H
4060 FOR I=1 TO 6: FOR J=1 TO 3
4070 IF E(I,J)<>0 THEN  GO SUB 9500: PRINT AT X,Y;S$(FN P(X,Y));
4080 NEXT J: NEXT I: GO TO 280
5000 REM ### TIPP BEKERDEZESE ##
5010 PRINT AT 19,0;"1.LAP SORA? ";
5020 LET K=3: GO SUB 6000
5030 LET I(1)=Q
5040 PRINT AT 21,0;B$
5050 PRINT AT 19,15;"1.LAP OSZLOPA? ";
5060 LET K=0: GO SUB 6000
5070 LET J(1)=Q: LET K=1: GO TO 7000
5080 GO SUB 3000
5090 PRINT AT 20,0;"2.LAP SORA? ";
5100 LET K=3: GO SUB 6000
5110 LET I(2)=Q
5130 PRINT AT 20,15;"2.LAP OSZLOPA? ";
5140 LET K=0: GO SUB 6000
5150 LET J(2)=Q: LET K=2: GO TO 7000
5160 GO SUB 3000
5170 IF I(1)<>I(2) OR J(1)<>J(2) THEN  RETURN
5180 PRINT AT 21,0;"KET KULONBOZOT KELL MEGADNOD! ";
5190 PRINT AT 20,0;B$;: GO TO 5090
6000 REM ### TIPP ELLENORZESE ##
6010 FOR I=1 TO 50: NEXT I: LET Q$=INKEY$
6020 LET ID=ID+1: IF ID>MI THEN  GO TO 9800
6030 LET Q$=INKEY$: IF Q$="" THEN  GO TO 6020
6040 IF Q$="A" OR Q$="a" THEN  GO TO 4000
6050 IF Q$<"1" OR CODE Q$>51+K THEN  GO TO 6020
6060 PRINT Q$;: LET Q=VAL Q$: RETURN
7000 REM ### NEMLETEZO LAP KISZURESE ###
7010 IF E(I(K),J(K))>0  THEN  GO TO 5000+K*80
7020 PRINT AT 21,0;"ITT MAR NINCS LAP!!!          ";
7030 PRINT AT 18+K,0;B$;: GO TO 4930+K*80
7500 REM ##### JATEKSZABALY ####
7510 CLS : PRINT "M E M O R Y    -   JATEKSZABALY"
7520 PRINT "A KEPERNYON 24 KARTYAT FOGSZ    LATNI, PERSZE LEFORD
ITVA. KET   KARTYA PART ALKOT, HA EGYFORMAK."
7530 PRINT "VAGY AZ EGYIKEN LEVO KIFEJEZES  ERTEKE EGYENLO A MAS
IKON LEVO   SZAMMAL. NEKED MINDIG KET KAR-  TYAT KELL EGYMAS UTA
N FELFORDI- "
```

```
7540 PRINT "TANOD, HA PART ALKOTNAK, MEGKA- POD OKET. ES 100-250
 PONTOT,   KULONBEN VISSZAFORDULNAK. AMENY-NYIVEL TOBBET PROBAL
GATSZ 12-NELANNYISZOR 50 PONTOT LEVONOK TO-"
7550 PRINT "LED, VISZONT HA MIND LEVESZED,  TOBB, MINT 500 JUTAL
OMPONTOT    KAPSZ. A KARTYAK EGY UVEGASZTA- LON VANNAK, S TE AZ
ASZTAL ALA"
7560 PRINT "NEZHETSZ, CSAK AZ A BETUT KELL  MEGNYOMNOD. A K. ILY
ENERT       K*K*300 PONTOT VONOK LE TOLED."
7570 RETURN
8000 REM ####### KEVERES #######
8010 FOR I=1 TO 18: READ Q$,V(I)
8020 LET X$=Q$(1 TO LEN Q$-1): LET Y$=Q$(LEN Q$)
8030 LET L=8-LEN X$
8040 LET V$(I)="        "(1 TO INT ((L+1)/2))+X$+"        "(1 TO INT (L/
2))+Y$: NEXT I
8050 FOR I=1 TO 5: LET U=1+INT (RND*18): LET V=1+INT (RND*18)
8060 LET Q$=V$(U): LET V$(U)=V$(V): LET V$(V)=Q$
8070 LET Q=V(U): LET V(U)=V(V): LET V(V)=Q
8080 NEXT I: LET PK=1+INT (RND*18)
8090 LET P=5+2*(RND<.5)+6*(RND<.5)
8100 FOR I=1 TO 6: FOR J=1 TO 3: LET PK=PK+P: IF PK>18 THEN  LET
 PK=PK-18
8110 LET R$(I,J)=V$(PK): LET E(I,J)=V(PK): NEXT J: NEXT I
8120 FOR I=1 TO 6: LET I=1+INT (RND*6): LET K=1+INT (RND*6): LET
 J=1+INT (RND*3): LET L=1+INT (RND*3)
8130 LET Q$=R$(I,J): LET R$(I,J)=R$(K,L): LET R$(K,L)=Q$
8140 LET Q=E(I,J): LET E(I,J)=E(K,L): LET E(K,L)=Q: NEXT I: RETU
RN
9000 REM ### I(K). SOR J(K). KARTYAJA    ###
9010 LET I=I(K): LET J=J(K)
9500 LET X=I*3-1: LET Y=J*10-9: RETURN
9600 REM ### HA MIND LEVETTE ###
9610 LET PO=INT (C4+(MI-ID)*C1): LET PO=PO+P0
9620 CLS : PRINT : PRINT : PRINT "GRATULALOK, MINDENT LE TUDTAL
               VENNI!"
9630 PRINT : PRINT : PRINT "EZERT ";P0;" JUTALOMONTOT KAPTAL.":
GO TO 9820
9800 REM ######## VEGE #########
9810 CLS : PRINT : PRINT : PRINT "SAJNOS, A JATEKIDOD LEJART!"
9820 LET PO=INT (PO-(PR-12)*(PR>12)): LET PO=PO*(PO>0)
9830 PRINT : PRINT : PRINT : PRINT "A JATEKBAN ELERT PONTSZAMOD:
               ";PO;" PONT."
```

A program felépítése

Mint látható, a program sokféle alkotóelemből áll össze. Maga a „főprogram" a következő részekből áll:

1. A konstansok inicializálásából — 40-es (a Sinclair Spectrumon 50-es) sortól.

2. A kártyafeliratokból — 140-es (a HT1080Z és a Commodore-64 esetén 120-as, a Sinclair Spectrum esetén 110-es) sortól.

3. A „játék kezdete" részből — 230-as (a HT1080Z és a Commodore-64 esetén 200-as, a Sinclair Spectrum esetén 180-as) sortól. Ez vezérli voltaképpen az egész játék lefolyását.

4. Két különböző „vége" részből:

a) ez abban az esetben működik, ha a játékos ideje lejár még mielőtt minden lapot levett volna, és a 11000-es, ill. a Sinclair Spectrum a 9800-as, a HT1080Z és a Commodore-64 esetén a 12000-es címen kezdődik.

b) ez akkor lép működésbe, ha az idő lejárta előtt sikerült a játékosnak minden lapot levennie és a 13000-es, a Sinclair Spectrum esetén a 9600-as, a HT1080Z és a Commodore-64 esetén 14000-es címnél indul.

A „vége" részek gépenként mások, ezért ezeket nem közöljük, viszont a programot igyekeztünk olyan áttekinthetően strukturálni, hogy minden rész eleje és vége könynyen megtalálható legyen benne.

Ezen az öt fő részen kívül a program nem kevesebb, mint 11 (a HT1080Z és a Sinclair Spectrum gépen 10) eljárást tartalmaz.

Ezek az eljárások a következők

1. A kártyákat „felrajzoló" rutin — 1000-es sortól.

2. A „Jó tipp megval." nevű rutin, amely két azonos kártya megfordítása utáni teendőket látja el — 2000-től.

3. Az egy-egy kártya megfordítását végző rutin — 3000-től.

4. Az „asztal alá nézés" rutinja — 4000-től.

5. A „tipp be" rutin, amely a felfordítandó kártyák adatainak lekérdezését irányító programrész — 5000-től.

6. A „tipp ellenőrzés" rutin, amely egy kártya egyik koordinátáját kérdezi meg és azt ellenőrzi — 6000-től.

7. A már nem létező — tehát levett — lapot kiszűrő, s azt újra megfordítani nem engedő rutin — 7000-től.

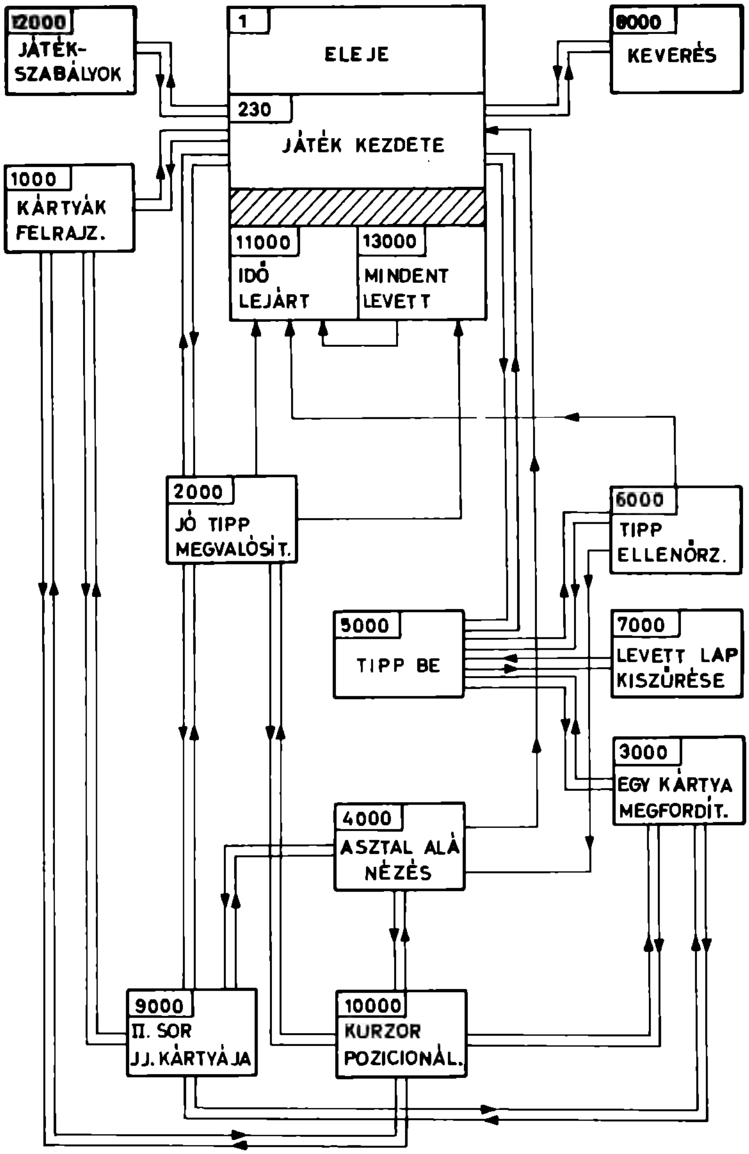
8. A kártya keverését végző rutin — 8000-től.

9. A kurzort a valahányadik sor valahányadik oszlopában levő kártya feliratának első karakterpozíciójára mozgató, ill. a HT1080Z-n és a Sinclair Spectrumon ennek a helynek a koordinátáit megadó rutin — 9000-től.

10. A kurzormozgató rutin — az M08X-en és a Commodore-64-en (utóbbinál még a 10-es sor is e rutin „része") — 10000-től.

11. A játékszabályokat kiíró rutin — 12000-től, ill. a Sinclair Spectrumon 7500-tól, a HT1080Z-n és a Commodore-64-en 13000-től.

A programról készült, meglehetősen bonyolultra sikerült, de így is csak vázlatos blokkdiagram a következő:



Megjegyzés: az ábrán a megfelelő M08X kezdőcímek láthatók.

A program működése

A főprogram

A program helyet biztosít a tömböknek. A két vektor csak segédváltozó, a keverés egyszerűsítése miatt használjuk. Az R$ mátrix tartalmazza az egyes kártyák feliratát és azonosítókódját egy-egy 9 karakteres stringben, amelynek utolsó karaktere az azonosítókód. Az E mátrix a kártyák pontértékét tartalmazza. MAXIDO (ill. a HT1080Z-n és a Sinclair Spectrumon MI) a maximális idő mértéke, amelynek a másodperchez való viszonyát csak hozzávetőlegesen lehet meghatározni, s beállítása tapasztalati úton történhet. Csak a játékos által felhasznált tiszta időt méri.

Az M08X gépen CC a képernyő kezdőcíme a tárban. A kételemű RS$ vektor (a Sinclair Spectrumon S$ (1)) a kártyák hátuljára írandó szöveget tartalmazza. Az IDB (egyes gépeken DB) a kártyák száma a képernyőn, C1 az a szorzó, amellyel a jutalompont számításakor a megtakarított időt szorozzuk, C2 a felesleges próbálkozásokért levonandó pontszám, C3 egy asztal alá nézés ára, C4 pedig az alap jutalompont teljes levétel esetén. Ezután (géptől függően!) néhány segédváltozónak adunk értéket, amelyeket a megjelenítésnél fognak használni. A Sinclair Spectrumon először le kell futtatni a karakterdefiniáló részt, s utána lehet beírni ezeket a sorokat, a grafikus üzemmód és a következő karakterek segítségével:

Q = ╭ ; A = ╰ ; F = |; G = –; P = ╮ ; L = ╯ .

Mivel a játékban egy párnak nemcsak két egyforma kártya számít, így a feliratok „bedatázása" egy kis körültekintést igényel. Minden egyes kártyának be kell írni a feliratát tetszőleges sorrendben, de az M08X-nél az egyedülálló kártya legyen az utolsó. Minden felirat végére kell írnunk még egy karaktert úgy, hogy ez a párba tartozó kártyák — és csak azok — esetén megegyezzen. Egy kártya felirata után a következő adat az, hogy a párjával együtt hány pontot ér.

A főprogram a kezdő szövegek (és esetleg a játékszabály) kiírása után hívja a 8000-es, majd az 1000-es, majd ezután egy látszólag végtelen ciklusban felváltva az 5000-es és a 2000-es rutint.

A 11000-nél (a HT1080Z és a Commodore-64 esetén a 10000-nél, a Sinclair Spectrum esetén a 9600-nál) kezdődő „vége" részben kiszámítja és kiírja a pontszámot, majd megáll.

A 13000-nél (a HT1080Z és a Commodore-64 esetén a 14000-nél, a Sinclair Spectrum esetén a 9600-nál) kezdődő „ha mind levette" részben kiszámítja és kiírja a jutalompontokat, majd elmegy a „vége" rutinnak az „idő lejárt" kiírását követő részére.

A rutinok

A rutinokat a használatba vételük sorrendjében ismertetjük. Az M08X-en és a Commodore-64-en a leggyakrabban használt rutin a 10000-es, amely az M08X-en a SOR értékének megfelelő sor OSZ-nak megfelelő pozíciójára, Commodore-64-en az X-edik sor Y-odik pozíciójára mozgatja a kurzort, tehát a következő PRINT utasítás ettől a pozíciótól kezdve fog kiírni.

Abszolút kisegítő jellegű a 9000-es rutin is. Érdekessége, hogy a kényelmesebb használat miatt a 9500-as sornál (az M08X kivételével) is van egy belépési pontja.

A 12000-es rutin (a HT1080-on és a Commodore-64-en 13000; a Sinclair Spectrumon 7500) egyszerűen kiírja a képernyőre a játékszabályokat. A 8000-es rutin beolvassa a kártyák feliratát és értékét, a feliratot azonosító karakter előtti részt 8 karakter hosszúságúra nyújtja, majd a szokott módszerrel (l. elrontott programok, 31—39. oldal) összekeverni a kártyákat.

Az 1000-es rutin felrajzolja a szükséges számú kártyát, beleírja a hátfeliratokat, s besorszámozza a sorokat és az oszlopokat. (A sorszámozás a Commodore-64 esetén helyhiány miatt elmaradt.) A rajzolás M08X-en POKE utasítással (inverzre

váltás), a többi gépen egyszerű PRINT utasítással történik. Persze a különböző képernyőméretek miatt más és más számú kártyák férnek el.

Az 5000-es rutin kiírja a négy kérdést (első kártya első koordinátájától a második kártya második koordinátájáig), mindegyik után hívja a 6000-es rutint, az onnan kapott számot elhelyezi, az első adat beírása után a régebbi szövegeket letörli, két-két adat lekérdezése után hívja a 7000-es vizsgálórutint (GOTO-val, s a visszatérés ON GOTO-val), majd a 3000-est. A két lap felfordítása után „megnézi", hogy két különböző lapot fordítottunk-e fel, ha nem, akkor üzenetet ír ki, s a 2. lapot (az M08X-en az elsőt is) újrakérdezi. Az M08X-en a dolog még annyival is bonyolultabb, hogy van egy egyedülálló lap. A vizsgálat során tehát azt is meg kell néznie a programnak, hogy a két azonos tipp nem erre a lapra vonatkozott-e. Ha igen, úgy nem kér új tippet, hanem leveszi ezt a lapot. A 6000-es rutinnál — míg vár egy karaktert — megindul az időmérés. Ha A betű a beadott karakter, akkor a 4000-es sorra megy, különben megvizsgálja, hogy megfelelő számjegyet kapott-e. Ha nem, akkor újra vár, ha igen, akkor azt kiírja, s visszatér.

A 7000-es rutinnál, ha a megadott helyen nincs lap, ezt üzenettel jelzi, és az illető lapot újrakéri.

A 3000-es rutin a megadott (II(k)-adik sor JJ(k)-adik pozícióján levő) kártyát felfordítja.

A 2000-es rutin 1-gyel növeli a próbálkozások számát, vár egy billentyű lenyomására, majd törli a szövegmezőt. Ha a két kártya nem alkot párt, akkor visszafordítja őket, ha pedig párt alkotnak, akkor mindkettőt törli a képernyőről és megnöveli a pontszámot, a pontmátrix megfelelő helyére pedig 0-t helyez (innen fogja ezentúl tudni, hogy itt nincs lap), majd a darabszámot csökkenti, és ha az 0 lett, akkor ugrik a 13000-es sorra.

Természetesen mindkét esetet (egyforma—nem egyforma) szöveggel is kijelzi. Az M08X gépen még van egy rész, amely a páratlan (pár nélküli) lap megtalálásakor lép működésbe.

A 4000-es rutin korrigálja a pontszámot, törli a szövegmezőt, ezután sorban felfordítja, majd visszafordítja a még asztalon levő kártyákat. Közben — az M08X kivételével — vár egy kicsit. (Igazság szerint oda is érdemes egy kis várakozást betenni, de az eredetiben nem volt, s nem akartuk azt megváltoztatni.)

Ezzel a program részeinek leírását befejeztük. A program némely része jól átvihető volt egyik gépről a másikra, de néhányat újra meg kellett írni (pl. az 1000-es rutin esetén). Itt előjött a gépek sok-sok előnye és hátránya, így érdemes a listákat összehasonlítani — melyik a legrövidebb?; legegyszerűbb?; legszebb?; legtöbbet tudó? stb. és miért? (Ugyanis az, hogy az M08X program a leghosszabb, pl. azon is múlik, hogy ott van egy pár nélküli kártya, amelyet külön le kell kezelni.) Egy szóval nem kívánjuk sugallni a véleményeket! Döntsön az olvasó!

8 TOTÓ: mit csinál ez az egysoros program?

A TOTÓ-hoz írt keretprogram biztosította, hogy azok a játékosok se tudják, milyen kérdéseket kapnak majd, akik játékra jelentkezésük előtt akár 10—15 versenyző játékát is végignézték, ugyanis készítettünk három teljes — egyenként 13+1 kérdést tartalmazó — sorozatot. A három sorozatban az azonos sorszámú kérdéseknél ugyanazokat a válaszvariációkat adtuk meg, de a hozzájuk tartozó programsorok már minden esetben mások voltak. Ily módon a helyes válasz mindig más és más lett. Keretprogramunk tehát minden versenyzőnél véletlenszerűen választotta ki a három sorozat valamelyikét, s ezen belül is a kérdéseket véletlenszerű sorrendben adta. Azért, hogy érthetőbb legyen mit is jelent az említett megoldás — vagyis hogy azonos válaszvariációknál különbözőek a programsorok —, nézzünk meg egy példát!

A kérdés az, hogy mit csinál a következő egysoros program? A válaszok a következők:

1. Kiírja, hogy 31
2. Kiírja, hogy 0
X. Mást ír ki

A TOTÓ egyik csomagjában ehhez a kérdéshez a következő programsor tartozik:

```
10 PRINT INT ((1+6)*(4+5)/2)
```

Ebben az esetben a helyes megoldás: 1.

A TOTÓ egy másik csomagjában ugyanehhez a kérdéshez a következő programsor jelenik meg:

```
10 PRINT INT (INT((1+6)/2)*(4+5))
```

Ebben az esetben természetesen a helyes megoldás már nem az 1-es. (Hadd ne áruljuk el, hogy 2 vagy X!)

Végül van egy harmadik TOTÓ-kollekció, amelyben ugyanehhez a kérdéshez tartozó programsor a következő:

```
10 PRINT INT ((1+6)*(INT((4+5)/2)-3.9))
```

Mondanunk sem kell, hogy nem volt könnyű 13 darab olyan kérdést összeállítani, amelyeknél lehetséges volt három egymástól alig különböző, ugyanakkor más és más eredményt adó programsort létrehozni. Mindezt ráadásul úgy, hogy ne is legyen túlságosan triviális a megoldás. (Két esetben ezt nem is tudtuk megoldani. Így a 2. és a 13. kérdés a szabályt erősítő kivétel lett.)

E TOTÓ-hoz készült keretprogramunkat csak az érdekesség és a teljesség kedvéért közöljük a megoldásoknál. Nem valószínű, hogy így, ebben a változatlan formában bárki is használná, ezért átírását sem tartottuk érdekesnek. Viszont közöljük a három teljes TOTÓ-sort. Ugyanazt láthatják tehát a következő listában, amit a versenyzők láthattak a képernyőjükön. Nekik persze a keretprogram az idejüket is mérte, és 4,5—5 perc után kegyetlenül abbahagyatta velük a TOTÓ-zást. Olvasónk nyugodtan gondolkodhat.

Hadd hívjuk föl még a figyelmet arra, hogy az igazi feladat az első 13 kérdés megválaszolása, hiszen a második és harmadik sorozatnál már segít az előző. Aki tehát

próbra akarja tenni magát, előbb oldja meg az első sorozatot, s csak utána nézze meg a másodikat, harmadikat.

(És mi van a +1-es kérdéssel? Az persze csak vicc. Úgy gondoltuk, aki eljut odáig hiszen a keretprogram csak annak adta meg a +1-es kérdést, aki az előzőekre hibátlanul válaszolt, az megérdemel egy munka nélküli pontot! Itt ugyanis minden választ helyesnek értékelt a számítógép!)

Íme tehát a TOTÓ három kérdéssorozata, amely az M08X gépre készült. Így a megoldásokhoz, a gondolkodáshoz ajánljuk olvasónk figyelmébe a 13. oldalon található — M08X BASIC különlegességeit összefoglaló — ismertetőnket. Ami pedig a másik három gépet illeti; a három TOTO-sorozat után azt is igyekeztünk összefoglalni (1. táblázat), hogy melyik kérdés melyik gépen értelmezhető és melyiken nem!

TOTÓ 1.

```
                                 +====================================+
                       1.        ! Mit csinal ez az egysoros program? !
                                 +====================================+
10 PRINT INT((1+6)*(4+5)/2)

1) KIIRJA, HOGY 31          2) KIIRJA, HOGY 0        X) MAST


                                 +====================================+
                       2.        ! Mit csinal ez az egysoros program? !
                                 +====================================+
15 IF INT(X)=FIX(X) THEN PRINT X

1) KIIRJA X ERTEKET, HA NEGATIV        2) KIIRJA X ERTEKET, HA NEM POZITIV
                        X) MAST


                                 +====================================+
                       3.        ! Mit csinal ez az egysoros program? !
                                 +====================================+
33 A$="3547":IF VAL(MID$(A$,3,1))=4 THEN A$=MID$(A$,2,1)

1) A$-NAK ERTEKET AD, S MIVEL A FELTETEL NEM TELJESUL, A KOVETKEZO SORRA MEGY
2) A$ ERTEKE "5" LESZ           X) A$ ERTEKE "3" LESZ


                                 +====================================+
                       4.        ! Mit csinal ez az egysoros program? !
                                 +====================================+
1 B=1:A=2:C=-1:PRINT A=B+C

1) 0-AT IR KI           2) -1-ET IR KI           X) HIBAT JELEZ


                                 +====================================+
                       5.        ! Mit csinal ez az egysoros program? !
                                 +====================================+
13 INPUT A,B:C=(A+B-ABS(A-B))/2:D=(A+B+ABS(A-B))/2:PRINT C,D
```

```
+====================================+
6.  ! Mit csinal ez az egysoros program? !
+====================================+
```

```
1 DATA 3,5,-1,3:FOR I=1 TO 4:READ K:S=S+K:NEXT I:PRINT S
```

1) 22-T IR KI   2) 10-ET IR KI   X) MAST

```
+====================================+
7.  ! Mit csinal ez az egysoros program? !
+====================================+
```

```
25 PRINT VAL(LEFT$(STR$(X),INSTR(STR$(X),".")))
```

1) KIIRJA X CSONKITOTT ERTEKET   2) KIIRJA EGESZ RESZET   X) MAST

```
+====================================+
8.  ! Mit csinal ez az egysoros program? !
+====================================+
```

```
0 PRINT A,"FOR I=1 TO 6:PRINT":I=I+1:NEXT I
```

1) NEXT WITHOUT FOR HIBAJELZEST AD   2) SYNTAX ERROR HIBAJELZEST AD   X) MAST

```
+====================================+
9.  ! Mit csinal ez az egysoros program? !
+====================================+
```

```
30 INPUT A,B:A=A+B:B=A+B:A=2*A:B=A-B:A=A/2:A=A-B
```

1) A ES B ERTEKET NEM VALTOZTATJA   2) MEGCSERELI A ES B ERTEKET   X) MAST

```
+====================================+
10.  ! Mit csinal ez az egysoros program? !
+====================================+
```

```
45 INPUT A$,B$:A$=A$+","+B$:B$=LEFT$(A$,INSTR(A$,",")-1):A$=RIGHT$(A$,LEN(A$)-LE
N(B$)-1)
```

1) A$ ES B$ ERTEKET NEM VALTOZTATJA   2) A$-T ES B$-T MEGCSERELI   X) MAST

```
+====================================+
11.  ! Mit csinal ez az egysoros program? !
+====================================+
```

```
5 INPUT X,Y:Z=SGN(X):X=Z*X+1:Z=SGN(Y)*Z:Y=SGN(Y)*Y-1:T=X*Y:PRINT Z*(T+X-Y-1)
```

1) KIIRJA ABS(X)*ABS(Y) ERTEKET   2) KIIRJA X*Y ERTEKET   X) MAST

```
+====================================+
12.  ! Mit csinal ez az egysoros program? !
+====================================+
```

```
60 A$="AKKORDKISERET":K=0:FOR I=1 TO LEN(A$):L=ASC(MID$(A$,I,1)):IF L<K THEN PRI
NT MID$(A$,1,I-1) ELSE K=L:NEXT I:PRINT A$
```

1)AZT IRJA KI, HOGY AKKORDKISERET   2) AZT IRJA KI, HOGY AKKOR   X) MAST

13. ! Mit csinal ez az egysoros program? !

```
10 INPUT P:IF P>1 AND P<46 AND P=INT(P) THEN FOR I=2 TO 7:IF P=INT(P/I)*I THEN P
RINT "NEM PRIM" ELSE NEXT I:PRINT "PRIM"
```

1) EGY SZAMROL ELDONTI, HOGY PRIM-E 2) EGY 2 ES 45 KOZTI EGESZ SZAMROL ELDONTI, HOGY PRIM-E X) MAST

13+1. ! Mit csinal ez az egysoros program? !

```
30 REM  GRATULALOK, HOGY KITALALTAD AZ ELSO 13-AT!
```

1) GRATULAL NEKED 2) GRATULAL A ZSURINEK X) SEMMIT

TOTÓ 2.

1. ! Mit csinal ez az egysoros program? !

```
10 PRINT INT(INT((1+6)/2)*(4+5))
```

1) KIIRJA, HOGY 31 2) KIIRJA, HOGY 0 X) MAST

2. ! Mit csinal ez az egysoros program? !

```
15 IF 0>X*SGN(X)*(SGN(X)-1) THEN PRINT X
```

1) KIIRJA X ERTEKET, HA NEGATIV 2) KIIRJA X ERTEKET, HA NEM POZITIV X) MAST

3. ! Mit csinal ez az egysoros program? !

```
33 B=3457:A$=STR$(B):IF VAL(MID$(A$,3,1))=4 THEN A$=MID$(A$,2,1)
```

1) A$-NAK ERTEKET AD, S MIVEL A FELTETEL NEM TELJESUL, A KOVETKEZO SORRA MEGY
2) A$ ERTEKE "5" LESZ X) A$ ERTEKE "3" LESZ

4. ! Mit csinal ez az egysoros program? !

```
1 PRINT A=TO+V1Z
```

1) 0-AT IR KI 2) -1-ET IR KI X) SYNTAX ERROR HIBAJELZEST AD

5. ! Mit csinal ez az egysoros program? !

```
13 INPUT A,B:C=(A-B+ABS(A-B))/2:D=(A-B-ABS(A-B))/2:PRINT C,D
```

1) KIIRJA ELOSZOR A ES B KOZUL A NAGYOBBIKAT, MAJD A KISEBBET
2) ELOSZOR IRJA KI A KISEBBET, UTANA A NAGYOBBAT
X) MAST

6. | Mit csinal ez az egysoros program? |

```
1 DATA 3,4,-1,5:FOR I=1 TO 4:READ K:S=S+K:NEXT I:PRINT S
```

1) 22-T IR KI   2) 10-ET IR KI   X) MAST

7. | Mit csinal ez az egysoros program? |

```
25 PRINT VAL(LEFT$(STR$(X),INSTR(STR$(X)+".",".")-1))-(X<>INT(X))*(X<0)
```

1) KIIRJA X CSONKITOTT ERTEKET   2) KIIRJA X EGESZ RESZET   X) MAST

8. | Mit csinal ez az egysoros program? |

```
0 PRINT A+"FOR I=1 TO 6:PRINT":I=I+1:GOTO 0
```

1) NEXT WITHOUT FOR HIBAJELZEST AD   2) SYNTAX ERROR HIBAJELZEST AD
X) MAST

9. | Mit csinal ez az egysoros program? |

```
30 INPUT A,B:A=A*B:B=A+B:C=B-A/(B-A)-1:A=C/(B-A-1)-1:B=C/(A+1)+1
```

1) A ES B ERTEKET NEM VALTOZTATJA   2) MEGCSERELI A ES B ERTEKET   X) MAST

10. | Mit csinal ez az egysoros program? |

```
45 INPUT A$,B$:A$=A$+B$:B$=B$+A$:A$=MID$(B$,LEN(B$)-LEN(A$)+1,2*LEN(A$)-LEN(B$))
:B$=LEFT$(B$,(LEN(B$)-LEN(A$))/2)
```

1) A$ ES B$ ERTEKET NEM VALTOZTATJA   2) A$-T ES B$-T MEGCSERELI   X) MAST

11. | Mit csinal ez az egysoros program? |

```
5 INPUT X,Y:Z=SGN(X):X=Z*X+1:Z=SGN(Y)*Z:Y=SGN(Y)*Y-1:Z=X*Y:PRINT Z+X-Y-1
```

1) KIIRJA ABS(X)*ABS(Y) ERTEKET   2) KIIRJA X*Y ERTEKET   X) MAST

12. | Mit csinal ez az egysoros program? |

```
60 A$="AKKORDKISERET":K=ASC(A$):FOR I=2 TO LEN(A$):L=ASC(MID$(A$,I,1)):IF L<K TH
EN PRINT LEFT$(A$,I-1) ELSE NEXT I:PRINT A$
```

1)AZT IRJA KI, HOGY AKKORDKISERET   2) AZT IRJA KI, HOGY AKKOR   X) MAST

13. | Mit csinal ez az egysoros program? |

```
10 INPUT P:IF P>1 AND P<46 AND P=INT(P) THEN PRINT MID$("NEM PRIM",(P<>2*INT(P/2
) OR P=2)*(P<>3*INT(P/3) OR P=3)*(P<>5*INT(P/5) OR P
=5)*(-4)+1)
```

1) EGY SZAMROL ELDONTI, HOGY PRIM-E   2) EGY 2 ES 45 KOZTI EGESZ SZAMROL ELDONTI, HOGY PRIM-E   X) MAST

```
                    +===================================+
        13+1.       ! Mit csinal ez az egysoros program? !
                    +===================================+
30 REM  GRATULALOK, HOGY KITALALTAD AZ ELSO 13-AT!

1) GRATULAL NEKED        2) GRATULAL A ZSURINEK        X) SEMMIT
```

TOTÓ 3.

```
                    +===================================+
        1.          ! Mit csinal ez az egysoros program? !
                    +===================================+
10 PRINT INT((1+6)*(INT((4+5)/2)-3.9))

1) KIIRJA, HOGY 31        2) KIIRJA, HOGY 0      X) MAST


                    +===================================+
        2.          ! Mit csinal ez az egysoros program? !
                    +===================================+
15 IF X<-SGN(X)-1 THEN PRINT X

1) KIIRJA X ERTEKET, HA NEGATIV        2) KIIRJA X ERTEKET, HA NEM POZITIV
                                X) MAST


                    +===================================+
        3.          ! Mit csinal ez az egysoros program? !
                    +===================================+
33 A$="35 47":IF VAL(MID$(A$,3,1))=4 THEN A$=MID$(A$,2,1)

1) A$-NAK ERTEKET AD, S MIVEL A FELTETEL NEM TELJESUL, A KOVETKEZO SORRA MEGY
2) A$ ERTEKE "5" LESZ             X) A$ ERTEKE "3" LESZ


                    +===================================+
        4.          ! Mit csinal ez az egysoros program? !
                    +===================================+
1 PRINT A$=B$+C$

1) 0-AT IR KI           2) -1-ET IR KI          X) HIBAT JELEZ


                    +===================================+
        5.          ! Mit csinal ez az egysoros program? !
                    +===================================+
13 INPUT A,B:C=(A+B+ABS(A-B))/2:D=(A+B-ABS(A-B))/2:PRINT C,D

1) KIIRJA ELOSZOR A ES B KOZUL A NAGYOBBIKAT, MAJD A KISEBBET
2) ELOSZOR IRJA KI A KISEBBET, UTANA A NAGYOBBAT
   X) MAST


                    +===================================+
        6.          ! Mit csinal ez az egysoros program? !
                    +===================================+
1 DATA 3,5,-1,15:FOR I=1TO 4:READ K:S=S+K:NEXT I:PRINT S

1) 22-T IR KI            2) 19-ET IR KI               X) MAST
```

```
                         +=====================================+
          7.             ! Mit csinal ez az egysoros program?  !
                         +=====================================+
25 PRINT VAL(LEFT$(STR$(X),INSTR(STR$(X)+".",".")-1))
```

1) KIIRJA X CSONKITOTT ERTEKET 2) KIIRJA X EGESZ RESZET X) MAST

```
                         +=====================================+
          8.             ! Mit csinal ez az egysoros program?  !
                         +=====================================+
1 PRINT VALTOZO,"FOR I=1 TO 6:PRINT":I=I+1:GOTO 0
```

1) NEXT WITHOUT FOR HIBAJELZEST AD 2) SYNTAX ERROR HIBAJELZEST AD X) MAST

```
                         +=====================================+
          9.             ! Mit csinal ez az egysoros program?  !
                         +=====================================+
30 INPUT A,B:A=A+B:B=A-B:A=A*(2*B-A):B=SQR(B*B-A):A=SQR(B*B+A)
```

1) A ES B ERTEKET NEM VALTOZTATJA 2) MEGCSERELI A ES B ERTEKET X) MAST

```
                         +=====================================+
          10.            ! Mit csinal ez az egysoros program?  !
                         +=====================================+
45 INPUT A$,B$:B$=B$+A$:A$=A$+B$:B$=MID$(A$,LEN(B$)+1):A$=RIGHT$(A$,(LEN(A$)-LEN
(B$))/2)
```

1) A$ ES B$ ERTEKET NEM VALTOZTATJA 2) A$-T ES B$-T MEGCSERELI X) MAST

```
                         +=====================================+
          11.            ! Mit csinal ez az egysoros program?  !
                         +=====================================+
5 INPUT X,Y:Z=SGN(X):X=Z*X+1:Z=SGN(Y)*Z:Y=SGN(Y)*Y-1:Z=X*Z*Y:PRINT Z+SGN(Z)*(X-Y
-1)
```

1) KIIRJA ABS(X)*ABS(Y) ERTEKET 2) KIIRJA X*Y ERTEKET X) MAST

```
                         +=====================================+
          12.            ! Mit csinal ez az egysoros program?  !
                         +=====================================+
60 A$="AKKORDKISERET":K=0:FOR I=1 TO LEN(A$):L=ASC(MID$(A$,I,1)):IF L>K THEN K=I
:NEXT I:PRINT MID$(A$,1,K) ELSE NEXT I:PRINT MID$(A$
,1,K)
```

1)AZT IRJA KI, HOGY AKKORDKISERET 2) AZT IRJA KI, HOGY AKKOR X) MAST

```
                         +=====================================+
          13.            ! Mit csinal ez az egysoros program?  !
                         +=====================================+
10 INPUT P:FOR I=2 TO SQR(INT(ABS(P))):IF P MOD I<>0 THEN NEXT I:PRINT "PRIM" EL
SE PRINT "NEM PRIM"
```

1) EGY SZAMROL ELDONTI, HOGY PRIM-E 2) EGY 2 ES 45 KOZTI EGESZ SZAMROL ELDONTI, HOGY PRIM-E X) MAST

```
                         +=====================================+
          13+1.          ! Mit csinal ez az egysoros program?  !
                         +=====================================+
30 REM  GRATULALOK, HOGY KITALALTAD AZ ELSO 13-AT!
```

1) GRATULAL NEKED 2) GRATULAL A ZSURINEK X) SEMMIT

1. táblázat

Összefoglaló táblázat a TOTÓ-kérdésekről, amelyekből kiderül, hogy melyik programsor értelmezhető a HT1080X, a Commodore-64 és a Sinclair Spectrum gépen, s ha nem értelmezhető, miért nem?

	HT1080Z	Commodore-64	Sinclair Spectrum *
1. kérdés			
TOTÓ 1.	rendben	rendben	rendben
TOTÓ 2.	rendben	rendben	rendben
TOTÓ 3.	rendben	rendben	rendben
2. kérdés			
TOTÓ 1.	rendben	A Commodore-64 és a Spectrum BASIC-ben nincs FIX(X), helyette a megoldás: SGN(X)*INT(ABS(X))	
TOTÓ 2.	rendben	rendben	rendben
TOTÓ 3.	rendben	rendben	rendben
3. kérdés			
TOTÓ 1.	rendben	rendben	A MID$ helyett indexelést kell alkalmazni!
TOTÓ 2.	rendben	rendben	A válasz X helyett 1 lesz, mivel a Spectrum az STR$ függvény végrehajtásakor nem rak a szám elé betűközt!
TOTÓ 3.	rendben	rendben	C Nonsense in BASIC hibajelzést kapunk, mivel a Spectrum az egy betűközt tartalmazó stringen nem tudja végrehajtani a VAL függvényt
4. kérdés			
TOTÓ 1.	rendben	rendben	rendben
TOTÓ 2.	rendben	rendben	A Spectrumon ez a kérdés a kulcsszavak egy billentyűről való beírása miatt nem megoldható
TOTÓ 3.	TM ERROR-t jelez; ha a sort így módosítjuk: A$ = (B$ + C$), akkor rendben!	rendben	A logikai érték különbsége miatt más eredményt kapunk, mint ami a megoldási variációkban szerepel
5. kérdés			
TOTÓ 1.	rendben	rendben	rendben
TOTÓ 2.	rendben	rendben	rendben
TOTÓ 3.	rendben	rendben	rendben
6. kérdés			
TOTÓ 1.	rendben	rendben	LET S = 0 kezdőértékadás szükséges
TOTÓ 2.	rendben	rendben	rendben
TOTÓ 3.	rendben	rendben	rendben
7. kérdés			
TOTÓ 1. TOTÓ 2. TOTÓ 3.	Az INSTR függvény hiánya miatt a programsor nem működik egyik gépen sem!		

```
                 +====================================+
       7.        ! Mit csinal ez az egysoros program? !
                 +====================================+
25 PRINT VAL(LEFT$(STR$(X),INSTR(STR$(X)+".",".")-1))
```

1) KIIRJA X CSONKITOTT ERTEKET   2) KIIRJA X EGESZ RESZET   X) MAST

```
                 +====================================+
       8.        ! Mit csinal ez az egysoros program? !
                 +====================================+
1 PRINT VALTOZO,"FOR I=1 TO 6:PRINT":I=I+1:GOTO 0
```

1) NEXT WITHOUT FOR HIBAJELZEST AD   2) SYNTAX ERROR HIBAJELZEST AD
X) MAST

```
                 +====================================+
       9.        ! Mit csinal ez az egysoros program? !
                 +====================================+
30 INPUT A,B:A=A+B:B=A-B:A=A*(2*B-A):B=SQR(B*B-A):A=SQR(B*B+A)
```

1) A ES B ERTEKET NEM VALTOZTATJA   2) MEGCSERELI A ES B ERTEKET   X) MAST

```
                 +====================================+
      10.        ! Mit csinal ez az egysoros program? !
                 +====================================+
45 INPUT A$,B$:B$=B$+A$:A$=A$+B$:B$=MID$(A$,LEN(B$)+1):A$=RIGHT$(A$,(LEN(A$)-LEN
(B$))/2)
```

1) A$ ES B$ ERTEKET NEM VALTOZTATJA   2) A$-T ES B$-T MEGCSERELI   X) MAST

```
                 +====================================+
      11.        ! Mit csinal ez az egysoros program? !
                 +====================================+
5 INPUT X,Y:Z=SGN(X):X=Z*X+1:Z=SGN(Y)*Z:Y=SGN(Y)*Y-1:Z=X*Z*Y:PRINT Z+SGN(Z)*(X-Y
-1)
```

1) KIIRJA ABS(X)*ABS(Y) ERTEKET   2) KIIRJA X*Y ERTEKET   X) MAST

```
                 +====================================+
      12.        ! Mit csinal ez az egysoros program? !
                 +====================================+
60 A$="AKKORDKISERET":K=0:FOR I=1 TO LEN(A$):L=ASC(MID$(A$,I,1)):IF L>K THEN K=I
:NEXT I:PRINT MID$(A$,1,K) ELSE NEXT I:PRINT MID$(A$
,1,K)
```

1)AZT IRJA KI, HOGY AKKORDKISERET   2) AZT IRJA KI, HOGY AKKOR   X) MAST

```
                 +====================================+
      13.        ! Mit csinal ez az egysoros program? !
                 +====================================+
10 INPUT P:FOR I=2 TO SQR(INT(ABS(P))):IF P MOD I<>0 THEN NEXT I:PRINT "PRIM" EL
SE PRINT "NEM PRIM"
```

1) EGY SZAMROL ELDONTI, HOGY PRIM-E   2) EGY 2 ES 45 KOZTI EGESZ SZAMROL
ELDONTI, HOGY PRIM-E   X) MAST

```
                 +====================================+
      13+1.      ! Mit csinal ez az egysoros program? !
                 +====================================+
30 REM  GRATULALOK, HOGY KITALALTAD AZ ELSO 13-AT!
```

1) GRATULAL NEKED   2) GRATULAL A ZSURINEK   X) SEMMIT

1. táblázat

Összefoglaló táblázat a TOTÓ-kérdésekről, amelyekből kiderül, hogy melyik programsor értelmezhető a HT1080X, a Commodore-64 és a Sinclair Spectrum gépen, a ha nem értelmezhető, miért nem?

	HT1080Z	Commodore-64	Sinclair Spectrum *
1. kérdés			
TOTÓ 1.	rendben	rendben	rendben
TOTÓ 2.	rendben	rendben	rendben
TOTÓ 3.	rendben	rendben	rendben
2. kérdés			
TOTÓ 1.	rendben	A Commodore-64 és a Spectrum BASIC-ben nincs FIX(X), helyette a megoldás: SGN(X)*INT(ABS(X))	
TOTÓ 2.	rendben	rendben	rendben
TOTÓ 3.	rendben	rendben	rendben
3. kérdés			
TOTÓ 1.	rendben	rendben	A MID$ helyett indexelést kell alkalmazni!
TOTÓ 2.	rendben	rendben	A válasz X helyett 1 lesz, mivel a Spectrum sz STR$ függvény végrehajtásakor nem rek a szám elé betűközt!
TOTÓ 3.	rendben	rendben	C Nonsense in BASIC hibajelzést kapunk, mivel a Spectrum az egy betűközt tartalmazó stringen nem tudja végrehajtani a VAL függvényt
4. kérdés			
TOTÓ 1.	rendben	rendben	rendben
TOTÓ 2.	rendben	rendben	A Spectrumon ez a kérdés a kulcsszavak egy billentyűről való beírása miatt nem megoldható
TOTÓ 3.	TM ERROR-t jelez; ha a sort így módosítjuk: A$ = (B$ + C$), akkor rendben!	rendben	A logikai érték különbsége miatt más eredményt kapunk, mint ami a megoldási variációkban szerepel
5. kérdés			
TOTÓ 1.	rendben	rendben	rendben
TOTÓ 2.	rendben	rendben	rendben
TOTÓ 3.	rendben	rendben	rendben
6. kérdés			
TOTÓ 1.	rendben	rendben	LET S = 0 kezdőértékadás szükséges
TOTÓ 2.	rendben	rendben	rendben
TOTÓ 3.	rendben	rendben	rendben
7. kérdés			
TOTÓ 1. TOTÓ 2. TOTÓ 3.	Az INSTR függvény hiánya miatt a programsor nem működik egyik gépen sem!		

	HT1080Z	Commodore-64	Sinclair Spectrum
8. kérdés			
TOTÓ 1.	rendben	rendben	A 0-s sorszám, valamint a kezdőértékadások hiánya miatt értelmetlen
TOTÓ 2.	rendben	rendben	
TOTÓ 3.	rendben	rendben	
9. kérdés			
TOTÓ 1.	rendben	rendben	rendben
TOTÓ 2.	rendben	rendben	rendben
TOTÓ 3.	rendben	rendben	rendben
10. kérdés			
TOTÓ 1.	Az INSTR függény hiánya miatt egyik gépen sem működne a sor		
TOTÓ 2.	rendben	rendben	A MID$ és a LEFT$ helyettesíthető a következőképpen: MID$ helyett: LET A$=B$ (LEN B$-LEN A$+1 TO LEN A$) LEFT$ helyett: LET B$=B$ (TO (LEN B$-LEN A$)/2)
TOTÓ 3.	rendben	rendben	A helyettesítések a következők lehetnek: MID$ helyett: LET B$=A$ (LEN B$+1 TO) RIGHT$ helyett: LET A$=A$(LEN A$+1-(LEN A$-LEN B$)/2 TO)
11. kérdés			
TOTÓ 1.	rendben	rendben	rendben
TOTÓ 2.	rendben	rendben	rendben
TOTÓ 3.	rendben	rendben	rendben
12. kérdés			
TOTÓ 1.	rendben	ELSE hiánya miatt nem megy	ELSE hiánya miatt nem megy
TOTÓ 2.	rendben	ELSE hiánya miatt nem megy	ELSE hiánya miatt nem megy
TOTÓ 3.	rendben	ELSE hiánya miatt nem megy	ELSE hiánya miatt nem megy
13. kérdés			
TOTÓ 1.	rendben	ELSE hiánya miatt nem megy	ELSE hiánya miatt nem megy
TOTÓ 2.	rendben	Attól eltekintve, hogy a sor csak két sorban fér el, rendben	−4 helyett +4 kell a programban és a MID$ helyett indexelés
TOTÓ 3.	P MOD I<>0 helyett P<>I INT(P/I) kell!		ELSE hiánya miatt nem megy!

A Spectrum BASIC szokásának megfelelően értékadásnál a LET utasítás szükséges. Ezt külön nem jelezzük, hiszen ez a Spectrumnál elkerülhetetlen.

1. A TOTÓ kérdéssor helyes válaszai:
   1, X, 2, 1, 2, 2, X, 1, 2, 2, 2, 2, X
2. A TOTÓ kérdéssor helyes válaszai:
   X, 1, X, X, X, X, 2, X, 1, 1, 1, X, 2
3. A TOTÓ kérdéssor helyes válaszai:
   2, 1, 1, 2, 1, 1, 1, 2, X, X, X, 1, X

A TOTÓ teljes programja az M08X gépre

```
10 ' ##### TOTO 1 #####
20 DEFSTR Q:Q=CHR$(34)
30 PRINT CHR$(17)
40 NSOR=14:TMAX=17000:T0=0:PONT1=0:C1=200
50 DIM A$(NSOR,2),TIP$(NSOR)
60 GOSUB 1000:PRINT CHR$(17)
70 FOR I=1 TO NSOR-1:READ TIP$(I):NEXT I:GOSUB 600
80 READ Q1,Q2,Q3,Q4,Q5
90 GOSUB 500
100 PRINT Q1:PRINT Q2:PRINT TAB(54);Q3:PRINT Q4:PRINT Q5
110 FOR I=1 TO NSOR:PRINT:PRINT:IF I<NSOR THEN PRINT STR$(I);". sor:":PRINT ELSE
 PRINT"13+1. sor:":PRINT
120 PRINT A$(V(I),1):PRINT:PRINT:PRINT A$(V(I),2)
130 PRINT:PRINT:PRINT:PRINT
140 PRINT "Tipp (1,2,x) ?   ";
150 T0=T0+1:IF T0>TMAX THEN 250 ELSE T$=INKEY$:IF T$="" THEN 150
160 PRINT T$:IF T$<>"1" AND T$<>"2" AND T$<>"x" AND T$<>"X" THEN 140
170 PRINT CHR$(17)
180 IF CHR$(ASC(T$)+32)=TIP$(V(I)) OR T$=TIP$(V(I)) OR I=14 THEN PRINT Q1:PRINT
Q2: PRINT "Az elozo kerdesre adott valaszod helyes!"
;TAB(54);Q3:GOTO 190 ELSE PRINT Q1:PRINT Q2:PRINT "Sajnos az elozo kerdesre ross
zul valaszoltal!";TAB(54);Q3:GOTO 200
190 PONT1=PONT1+C1
200 PRINT Q4:PRINT Q5
210 IF I=NSOR-1 AND PONT1<C1*(NSOR-1) THEN 230 ELSE NEXT I
220 PRINT:PRINT:PRINT:PRINT
230 PRINT "Toto osszpontszama=";PONT1:PONT=PONT+PONT1
240 END
250 PRINT:PRINT "Sajnos lejart az idod.":PRINT: GOTO 230
470 '
480 ' ##### KEVERO #####
490 '
500 DIM V(NSOR):RANDOMIZE T0
510 FOR I = 1 TO NSOR:V(I)=I:NEXT I
520 N9=6:N8=NSOR-1-N9::FOR I=1 TO 15:A=INT(RND*N9)+1:B=INT(RND*N9)+1
530 J=V(A):V(A)=V(B):V(B)=J:NEXT I
540 FOR I=1 TO 10:A=7+INT(RND*N8):B=7+INT(RND*N8):SWAP V(A),V(B):NEXT I:RETURN
570 '
580 ' ##### ADATOK #####
590 '
600 A$(1,1)="10 PRINT INT((1+6)*(4+5)/2)":A$(1,2)="1) KIIRJA, HOGY 31        2) K
IIRJA, HOGY 0      X) MAST"
610 DATA 1
620 A$(2,1)="15 IF INT(X)=FIX(X) THEN PRINT X":A$(2,2)="1) KIIRJA X ERTEKET, HA
NEGATIV        2) KIIRJA X ERTEKET, HA NEM POZITIV
                                X) MAST"
630 DATA x
640 A$(3,1)="33 A$="+CHR$(34)+"3547"+CHR$(34)+":IF VAL(MID$(A$,3,1))=4 THEN A$=M
ID$(A$,2,1)":A$(3,2)="1) A$-NAK ERTEKET AD, S MIVEL
A FELTETEL NEM TELJESUL, A KOVETKEZO SORRA MEGY    2) A$ ERTEKE "+CHR$(34)+"5"+CH
R$(34)+" LESZ             "
650 AAAA$="X) A$ ERTEKE "+CHR$(34)+"3"+CHR$(34)+" LESZ":A$(3,2)=A$(3,2)+AAAA$
660 DATA 2
670 A$(4,1)="1 B=1:A=2:C=-1:PRINT A=B+C":A$(4,2)="1) 0-AT IR KI            2) -1-ET
 IR KI            X) HIBAT JELEZ"
```

```
680 DATA 1
690 A$(5,1)="13 INPUT A,B:C=(A+B-ABS(A-B))/2:D=(A+B+ABS(A-B))/2:PRINT C,D":A$(5,
2)="1) KIIRJA ELOSZOR A ES B KOZUL A NAGYOBBIKAT, MA
JD A KISEBBET                   2) ELOSZOR IRJA KI A KISEBBET, UTANA A NAGYOBBAT
      X) MAST"
700 DATA 2
710 A$(11,1)="5 INPUT X,Y:Z=SGN(X):X=Z*X+1:Z=SGN(Y)*Z:Y=SGN(Y)*Y-1:T=X*Y:PRINT Z
*(T+X-Y-1)":A$(11,2)="1) KIIRJA ABS(X)*ABS(Y) ERTEKE
T        2) KIIRJA X*Y ERTEKET      X) MAST"
720 DATA 2
730 A$(7,1)="25 PRINT VAL(LEFT$(STR$(X),INSTR(STR$(X),"+CHR$(34)+"."+CHR$(34)+")
))":A$(7,2)="1) KIIRJA X CSONKITOTT ERTEKET      2) K
IIRJA X EGESZ RESZET     X) MAST"
740 DATA x
750 A$(8,1)="0 PRINT A,"+Q+"FOR I=1 TO 6:PRINT"+Q+":I=I+1:NEXT I":A$(8,2)="1) NE
XT WITHOUT FOR HIBAJELZEST AD          2) SYNTAX ERROR
 HIBAJELZEST AD                                               X) MAST"
760 DATA 1
770 A$(9,1)="30 INPUT A,B:A=A+B:B=A+B:A=2*A:B=A-B:A=A/2:A=A-B":A$(9,2)="1) A ES
B ERTEKET NEM VALTOZTATJA        2) MEGCSERELI A ES B
ERTEKET       X) MAST"
780 DATA 2
790 A$(10,1)="45 INPUT A$,B$:A$=A$+"+Q+","+Q+"+B$:B$=LEFT$(A$,INSTR(A$,"+Q+","+Q
+")-1):A$=RIGHT$(A$,LEN(A$)-LEN(B$)-1)":A$(10,2)="1)
 A$ ES B$ ERTEKET NEM VALTOZTATJA        2) A$-T ES B$-T MEGCSERELI       X) MAST"
800 DATA 2
810 A$(6,1)="1 DATA 3,5,-1,3:FOR I=1 TO 4:READ K:S=S+K:NEXT I:PRINT S":A$(6,2)="
1) 22-T IR KI              2) 10-ET IR KI
  X) MAST"
820 DATA 2
830 A$(12,1)="60 A$="+Q+"AKKORDKISERET"+Q+":K=0:FOR I=1 TO LEN(A$):L=ASC(MID$(A$
,I,1)):IF L<K THEN PRINT MID$(A$,1,I-1) ELSE K=L:NEX
T I:PRINT A$":A$(12,2)="1)AZT IRJA KI, HOGY AKKORDKISERET      2) AZT IRJA KI, HO
GY AKKOR     X) MAST"
840 DATA 2
850 A$(13,1)="10 INPUT P:IF P>1 AND P<46 AND P=INT(P) THEN FOR I=2 TO 7:IF P=INT
(P/I)*I THEN PRINT "+Q+"NEM PRIM"+Q+" ELSE NEXT I:PR
INT "+Q+"PRIM"+Q+""
860 DATA x
870 A$(13,2)="1) EGY SZAMROL ELDONTI, HOGY PRIM-E        2) EGY 2 ES 45 KOZTI EGES
Z SZAMROL      ELDONTI, HOGY PRIM-E                  X) M
AST"
880 A$(14,1)="30 REM  GRATULALOK, HOGY KITALALTAD AZ ELSO 13-AT!":A$(14,2)="1) G
RATULAL NEKED        2) GRATULAL A ZSURINEK          X
) SEMMIT"
890 RETURN
900 DATA "                                                    :        T O T O"
,"
  :"
910 DATA                                          ":   Mit  csinal  ez  az","
:   egysoros  program?"
920 DATA "                                              +----------------
----------"
970 '
980 ' ##### SZOVEG #####
990 '
1000 PRINT CHR$(17):PRINT:PRINT:PRINT"        "+Q+"Mit csinal ez az egysoros prog
ram?"+Q
1010 PRINT:PRINT"                         T O T O ":PRINT:PRINT:PRINT
1020 PRINT"Kapsz egymas utan 13 kulonbozo programsort.":PRINT:PRINT"Mindegyik so
r utan haromfele megoldasbol valaszthatsz.":PRINT
1030 PRINT"(Ezeket 1), 2) es X) jeloli.)":PRINT:PRINT"Helyes valaszaid 200 - 200
  pontot ernek.":PRINT:PRINT"Ha negy es fel percen b
elul mind a 13 kerdesre helyes valaszt adsz,":PRINT:PRINT"megkapod a 13+1. kerde
st is.":PRINT
1040 PRINT"Vigyazz, az ido nagyon keves, 1 - 1  kerdesre csak kb. 15 masodperc j
```

```
ut.":PRINT:PRINT:PRINT:PRINT"HA ELOLVASTAD, NYOMD ME
G A BETUKOZ BILLENTYUT!";
1050 O$=INKEY$
1060 T0=T0+1:IF T0>TMAX THEN 250 ELSE T$=INKEY$:IF T$="" THEN 1060
1070 RETURN
```

9 Hibakeresés

Egy rövid program jelent meg a játékosok előtt a képernyőn. Ebben a programban kellett megkeresniük azt a sort, amelyben egy nyilvánvaló hibát hagytunk. Ennek a sornak a sorszámát kellett megoldásként közölni. Ezidáig ebben semmi különösebben érdekes, szellemes sincs. Azért, hogy legyen, kitaláltuk hát, hogy a program, amely megjelent a képernyőn, voltaképpen önmaga volt az a program (a DATA részt kivéve), amely kiírta a képernyőre a programot. Így a program kiválasztott a saját sorai közül egyet, s annak helyére az előre megadott hibás sorok közül kiírta a megfelelőt. Végül megkérdezte, hogy melyik a hibás sor, és várta a bemenő adatot. A program tehát önmagával játszott. Sok versenyzőnk észrevette a „trükköt", s jót nevetett rajta, a hibákat viszont jóval kevesebben találták meg.

Az eddigiekből az is kiderült, hogy ahány sora volt a programunknak, annyiféle hibalehetőség volt, mindegyik sornak írtunk ui. egy hibás változatot.

Ehelyütt úgy gondoltuk, e játékból bőven elég, ha ízelítőnek közlünk egyetlen listát egyetlen hibás sorral. Ha túl könnyen találják meg a hibát, ne feledjék, hogy ott a BNV-n erre a feladatra nem maradt több idejük a játékosoknak 1,5—2 percnél!

```
10 N=15:TMAX=4800:DIM JO$(N),ROS$(N)
15 FOR I=1 TO N:READ JO$(I),ROS$(I):NEXT I
20 READ KERDES$,SAJNOS$,LEJAR$,MEHET$,OK$,PONTS$:PRINT MEHET$
25 L=L+1:T=T+1:IF T>TMAX THEN 70 ELSE C$=INKEY$
30 IF C$=URES$ THEN 25 ELSE RANDOMIZE L
35 L=INT(RND*(N))+1:JO$(L)=ROS$(L):FOR I=1 TO N:PRINT JO$(I):NEXT I:PRINT
40 PRINT KERDES$:NK=0
45 C$=INKEY$:T=T+1:IF T>TMAX THEN 70 ELSE IF C$=URES$ THEN 45
50 IF C$=CHR$(13) THEN 55 ELSE NK=NK*10+VAL(C$):PRINT C$;:GOTO 45
55 PRINT:IF NK=5*L+5 THEN 65 ELSE IF NK<0 VAGY NK>N*5+5 THEN 40
60 PRINT SAJNOS$:NTIP=NTIP+1:PONT0=PONT0-200:GOTO 40
65 PONT0=PONT0+800:PRINT OK$:GOTO 75
70 PRINT LEJAR$
75 PONT0=INT(PONT0+(TMAX-T)/40)
80 PRINT PONTS$;PONT0:END

Melyik sor hibas?
```

Megoldás Az 55-ös sorban van a hiba. Az OR helyett VAGY vagyon írva, amit ez az „ánglius" BASIC nem ért!

A vetélkedő döntőjében ötféle feladatban mérhették össze tudásukat a versenyzők. Az öt feladat közül az egyiket telefonon kapták meg, másfél nappal a döntő előtt. Erre az ún. házifeladatra később térünk vissza. Kezdjük a döntő feladatsorát a legösszetettebb és talán legnehezebb feladattal, amely ráadásul két részből áll.

Rejtvényfejtő olvasóink ennél a feladatnál lesznek a legnehezebb helyzetben. Választaniuk kell ui., hogy a két feladat közül melyiket oldják meg, hiszen a versenyzők a feladat első részében készen kaptak egy játékprogramot a megfelelő játékszabállyal, s megadott idő alatt minél eredményesebben kellett ezt a játékot lejátszaniuk. Önkritikusan be kell vallanunk, hogy itt mi, a vetélkedő szervezői, elkövettünk egy hibát. Könnyebbnek becsültük a játékot, mint amilyen valójában volt. Ily módon kevés felkészülési időt és játékidőt adtunk rá, aminek az lett a következménye, hogy a megszerezhető 160 pontból mindössze 10-et sikerült a játékosoknak elérniük. (Persze ebben azért a játékosok is ludasak voltak, amennyiben nem olvasták el elég figyelmesen a kicsit bonyolult játékszabályt!) A döntőben részt vevők következő feladata ezután ennek a játékprogramnak a megírása volt.

Ha tehát olvasónk a játékot akarja kipróbálni, úgy be kell írnia gépébe a játék programját. Ezután már nem oldhatja meg a feladat második részét, hiszen nem írhatja meg önállóan a programot. Olvasónk persze kezdhetné fordítva is a dolgot, vagyis először megírhatná a programot, s utána játszhatna vele. Csakhogy ez esetben a képzeletére van bízva a program működése, ill. mindent az itt olvasható játékszabályból kell kiolvasnia. Ez pedig nem könnyű feladat. A legjárhatóbb út talán, ha olvasónk valakit megkér a program gépbe írására. Ezután lejátszhatja a játékot, s megírhatja a saját programját. Kezdjük mindenképpen a játék szabályának ismertetésével. Szóról szóra ugyanazt a szabályt közöljük, amelyet a döntőbe jutott 16 versenyző is kézhez kapott a BNV-n.

Játékszabály

SCITELtown-ba, az őserdei SZKIta faluba érkezik John SCILful, a világhírű etnográfus. Mivel többek közt a SZKIták családi életét is szeretné tanulmányozni, meg akarja tudni, hogy a faluban élő 26 házaspár: 26 férfi és 26 nő közül ki kinek a férje, ill. a felesége. Igen ám, de a SZKIta erkölcsök tiltják a párkapcsolatok nyílt felfedését, tehát a bennszülöttek nem mondhatják meg, hogy ki kinek a párja. De mert az etnográfusnak is szeretnének segíteni, meg játszani is igen szeretnek, a következőt ajánlják SCILfulnak:

Kimegy a falu minden felnőtt lakója, tehát 26 férfi és 26 nő a falu főterére. SCILful a főtér egyik oldalára tetszés szerinti csoportokban felállítja a férfiakat. Ezután a főtér másik oldalán a férfiak csoportjainak megfelelően felállnak a nők is. Tehát ha például a 26 férfi a következő létszámú csoportokban áll: 1-2-3-4-5-5-6, akkor a nők is ilyen létszámú csoportokban fognak állni, s ráadásul az egymásnak megfelelő létszámú csoportokban az egymásnak megfelelő férjek és feleségek fognak állni. Tehát a 4 fős nőcsoportban a 4 fős férficsoport feleségei állnak. De hogy a négy férfinak a négy nő közül melyik is a felesége, ezt nem árulják el. Sőt, az azonos létszámú két csoport, jelen esetben a két ötös csoport tovább bonyolítja a helyzetet. Ugyanis azt sem árulják el, hogy melyik ötös női csoport melyik ötös férficsoporthoz tartozik. Az azonban

biztos, hogy bármelyik két nő azonos csoportban áll, ha a férjeik is azonos csoportban állnak, de különbözőben, ha a férjeik is különbözőben állnak.

SCILful ezután átrendezheti a férfiak csoportjait, mire a nők is átrendezik saját csoportjaikat az előző szabályok szerint. Ám SCILfulnak minden átrendezés után 26 darab M08X BASIC könyvet kell fizetségül átadnia a törzsfőnöknek. Amikor SCILful úgy érzi, hogy már tudja, hogy ki kinek a férje, ill. a felesége, akkor közölheti ezt a törzsfőnökkel, s hogy ő se járjon rosszul a játékban, a törzsfőnök megígéri neki, hogy minden helyes tippjéért megajándékozza őt a SZKIták kedvenc csemegéjével, egy-egy csomag aPROPEReccel.

Az M08X gép segítségével SCILful helyébe képzelhetjük magunkat. A gépen a férfiakat és a nőket betűk jelölik. A 26 betű a következő: abcdefghijklmnopqrstuvwxyz, ezek — mármint a kisbetűk — jelölik a férfiakat. A nagybetűk — ugyanezek a betűk — jelölik a nőket. A gép kérdésére, hogy:

„Kérem a férfiak csoportjait"

írjuk be a kisbetűket a kívánt csoportosításban. Minden csoport után hagyjunk egy darab space-t. Tehát például egy beírás a következő lehet:

a bc def ghij klmno pqrst uvwxyz

Erre majd a gép válaszul kiírja a nők megfelelő csoportjait, például így:

W GH KOP ADUM NBXZE FIYRC JLQSTV

Ebből azt pl. máris tudjuk, hogy „a" felesége: „W", hogy „b" felesége: „G" vagy „H" stb. De vigyázzunk! A két ötös csoport nem biztos, hogy egymásnak megfelelője. Tehát „k" felesége lehet hogy az NBXZE csoportban van, de lehet hogy az FIYRC-ben. Ez az azonos létszámú csoportok átka.

Ezután a gép megkérdezi, hogy:

„Akarsz-e tippelni?"

Ha nemmel válaszolunk, újra kérdezi a férfiak csoportjait!

Apróság, de fontos: vigyázzunk, mert a férfiak csoportjainak begépelésekor javítani nem lehet! Tehát az egyszer beírt betűt a gép már nem tudja módosítani! (Természetesen a játék során papírt, ceruzát szabad használni.)

Nem fogad el a gép olyan csoportosítást, ill. beírást sem, amelyben egy betű kétszer szerepel! Épp ezért, ha valaki téveszt a beírás során, és újra akarja kezdeni annak a csoportosításnak a beírását, ha észrevette a hibáját, írjon be egy olyan betűt, amit már egyszer beírt, erre a gép kiírja, hogy mondjuk az „a" betű kétszer szerepel, s újra várja a beírást. Az ilyen sorokat nem számolja a gép.

Természetesen nem fogad el a gép olyan karaktert sem, amely nem a 26 betű valamelyike. Könnyítés viszont, hogy nem kötelező mindig mind a 26 betűt használni, tehát megadható olyan csoportosítás is, amelyben kevesebb „férfit" állítunk fel a főtéren.

A beírás végén mindig CR-rel kell elküldeni a sort, mire rövid számolgatás után a képernyőn megjelenik a válasz.

A játékra összesen tíz perc van. A kilencedik perc végén, ha addig nem adtuk be tippjeinket, a gép már nem fogad el további csoportosításokat, hanem csak a tippeket. (Ezt az időt itt a könyvünkben közölt programoknál megváltoztattuk. Kilencedik perc helyett már a nyolcadik perc végén kell tippelni.) A tippelés során a gép kiírja a nők betűjét — tehát a nagybetűket —, s kéri hozzájuk a megfelelő férfiak kisbetűit. Tippelni csak egyszer lehet és egyszerre! Nincs tehát lehetőség többszöri vagy több részletben való tippelésre!

Pontozás: A játékkal összesen 160 aPROPERec, azaz pont szerezhető. Egy rossz tipp 6 pont levonást jelent. A játéknak van egy optimális megoldása, azaz van egy minimális számú csoportosítás, amire szükség van a megoldáshoz. E minimális szám fölött minden újabb csoportosítás mínusz pontokat jelent, mégpedig 20—20-at. A játékra 15 perc felkészülési idő van.

Apró eltérések az M08X és a másik három gép játékprogramja között:

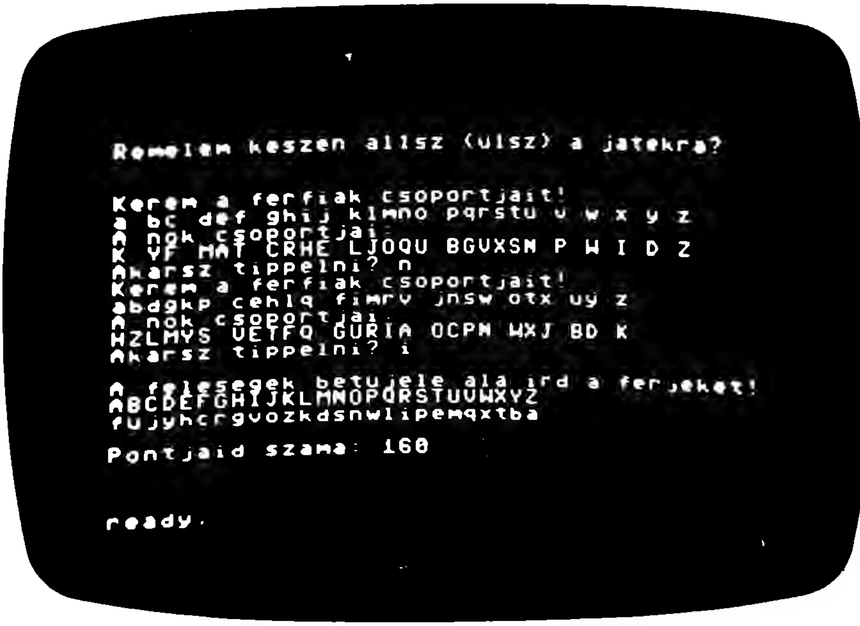
1. A HT1080Z gépen technikai okokból a kis- és nagybetűk használata éppen fordítottja a szabályban levőknek. Tehát a férfiak csoportjait kell nagybetűkkel beírnunk, s a gép a nőket jelöli kisbetűkkel.

2. Figyelem! A HT1080Z gépnél a játékhoz szükség van a tárbővítésre!

3. A beírás során is tapasztalunk bizonyos különbségeket az egyes gépeknél. A betűközbillentyű például „nyomot hagy" a Sinclair Spectrumon, éspedig aláhúzásjelet, a HT-nél pedig kurzor jelzi, hogy hol vagyunk.

Megoldás

Bármilyen hihetetlen is, összesen kétszeri csoportba állítás után valamennyi párt ki lehet találni! Bizonyítja ezt az alábbi képernyőfotó.



36. képernyőkép
SCILful és a SZKIták

A SZKIták bosszúja: a programírási feladat

Ez volt a játékprogrammal kapcsolatos második nagy feladat: egyszerűen meg kellett írni azt a programot, amellyel ez a játék lejátszható. Tekintettel azonban arra, hogy ott a BNV-n a versenyzők 40 perc papír melletti munkaidőt és 40 perc gépidőt kaptak lehetőségük volt arra, hogy bizonyos dolgokat készen kapjanak a programhoz. **Megkaphatták azt a szubrutint, amely** a betűk keverését végzi (tehát pl. a voltaképpeni **párosítást előállítja**).

Kihagyhatták ezenkívül programjukból azt a vizsgálórészt, amely a beírt betűk vizsgálatát végzi. Tehát feltételezhették, hogy a játékos nem követ el hibát, s ezért nem kell a programnak vizsgálnia, hogy valóban kisbetűket – ill. a HT1080Z-n nagybetűket — írt-e be, s hogy nem szerepeltet-e többször egy betűt.

Mindezek a könnyítések ott a döntőben lehetőségek voltak. Akadt versenyző, aki a keverőrutint is megírta; akadt, aki jóformán semeddig sem jutott a feladat megoldásában.

Önnek, kedves olvasó, nincs engedmény. Ha megírja a programot, mert próbára akarja tenni magát, ám lásson neki. Előre is jelezzük, hogy nem kis dolog, amire vállalkozik. Ha készen van, vesse össze programját a mienkkel, amely itt következik négy változatban! S ha az összevetés eredményeképpen úgy veszi észre, hogy az Öné jobbra, szebbre és főleg egyszerűbbre sikerült, úgy méltán büszke lehet magára.

```
1 '##### SCILful es a SZKItak #####
5 DEFSTR S,Q:DEFINT H,J-L,P
10 DIM H(26),F$(26),N$(26),SE(52)
20 MI=36000!:C=.8:PO=4
30 PRINT CHR$(17):PRINT:PRINT "       S C I L f u l   e s   a   S Z K I t a k
   -   j a t e k ":PRINT:PRINT
40 Q=INKEY$:PRINT "Remelem keszen allsz (ulsz) a jatekra? ";
50 FOR I=1 TO 26:S=S+CHR$(64+I):NEXT I:NJ$=S
60 ID=ID+1:Q=INKEY$:IF Q="" OR Q="n" THEN 60
70 PRINT Q:RANDOMIZE ID:GOSUB 5555:NO$=K$
80 PRINT:PRINT "Kerem a ferfiak csoportjait!":Q=INKEY$:V$="":CS=1:F$(CS)=""
90 ID=ID+1:IF ID>C*MI THEN 4000
100 Q=INKEY$:IF Q="" THEN 90
110 IF ASC(Q)=13 THEN 150
120 IF (Q<"a" OR Q>"z") AND Q<>" " THEN PRINT TAB(60);"Hibas karakter: ";Q:GOTO
80
125 PRINT Q$;
130 IF Q$=" " THEN CS=CS+1:F$(CS)="":GOTO 90
135 FOR J=1 TO CS:IF INSTR(F$(J),Q)<>0 THEN 145
140 NEXT J:F$(CS)=F$(CS)+Q:GOTO 90
145 PRINT:PRINT "A '";Q;"' karakter ketszer szerepel!":GOTO 80
150 I=0:PRINT TAB(60);"OK."
160 I=I+1
165 IF F$(I)="" THEN CS=CS-1:FOR J=I TO CS:F$(J)=F$(J+1):NEXT J:IF I<CS THEN 165
170 H(I)=LEN(F$(I)):IF I<CS THEN 160
175 IF F$(CS)="" THEN CS=CS-1
180 IF CS=0 THEN 80
230 FOR I=1 TO CS:N$(I)=""
240 FOR J=1 TO H(I):N$(I)=N$(I)+MID$(NO$,ASC(MID$(F$(I),J,1))-96,1):NEXT J
250 S$=N$(I):GOSUB 5555:N$(I)=K$:NEXT I
260 FOR I=1 TO 20:L1=1+INT(RND(1)*CS):L2=1+INT(RND(1)*CS)
270 IF H(L1)=H(L2) THEN SWAP N$(L1),N$(L2)
280 NEXT I:PRINT "A nok csoportjai:"
290 FOR I=1 TO CS:PRINT N$(I)+" ";:NEXT I
300 PRINT:PRINT "Akarsz tippelni? ";:Q=INKEY$
310 ID=ID+1:IF ID>C*MI THEN 4000
320 Q=INKEY$:IF Q="" THEN 310
330 PRINT Q$:IF Q$="n" THEN PR=PR+1:GOTO 80
990 '
1000 '#################### TIPPELES #######################
1010 PRINT:PRINT "A felesegek betujele ala ird a ferjeket!"
1020 PRINT NJ$:ZZ=0:Q=INKEY$
1030 ID=ID+1:IF ID>MI THEN 5000
1040 Q=INKEY$:IF Q="" THEN 1030
1050 IF Q<"a" OR Q>"z" THEN PRINT:PRINT "Hibas karakter: ";Q:PO=4:GOTO 1010
1060 ZZ=ZZ+1:PRINT Q;:QQ=MID$(NO$,ASC(Q)-96,1)
1070 IF QQ=MID$(NJ$,ZZ,1) THEN PO=PO+6
1080 IF ZZ=26 THEN 5020
1090 GOTO 1030
3990 '
4000 '############## KOTELEZO TIPPELES ####################
4010 PRINT:PRINT "Mar csak az ido";(1-C)*100;"%-a van hatra, most tippelni kell.
"
4020 GOTO 1000
4990 '
```

```
5000 '######################## VEGE ############################
5010 PRINT:PRINT:PRINT "Sajnos lejart az idod!":PRINT
5020 PO=PO+(PR>1)*(PR-1)*20:PO=-INT(PO)*(PO>0)
5030 PRINT:PRINT:PRINT "Pontjaid szama:";PO
5040 PRINT:PRINT:PRINT:END
5554 '
5555 '############### STRING KEVERO RUTIN #################
5560 N=LEN(S):FOR H=1 TO N:SE(H)=MID$(S,H,1):NEXT H
5570 FOR H=1 TO 5:L1=1+INT(RND(1)*N):L2=1+INT(RND(1)*N)
5580 SWAP SE(L1),SE(L2):NEXT H
5590 PK=1+INT(RND(1)*N)
5600 P=INT(RND(1)*3)*6+5-2*(RND(1)<.5):IF N=P*INT(N/P) THEN 5600
5610 K$="":FOR H=1 TO N:PK=PK+P
5620 IF PK>N THEN PK=PK-N:GOTO 5620
5630 K$=K$+SE$(PK):NEXT H:RETURN
```

```
1 '##### SCILful és a SZKIták - játék #####
5 CLEAR 1000:DEFSTR Q,S:DEFINT H,I-L,P
10 DIMH(26),F¤(26),N¤(26),SE(52),F1¤(52)
20 MI=19000:C=.8:PO=4
30 CLS:PRINT:PRINT:PRINT "S C I L f u l   é s   a   S Z K I t á k   -   j á
 t é k":PRINT:PRINT
40 Q=INKEY¤:PRINT "Remélem készen állsz (ülsz) a játékra? ";
50 FOR I=97 TO 122:S=S+CHR¤(I):NEXT I:NJ¤=S
60 IDO=IDO+1:Q=INKEY¤:IF Q="" OR Q="N" THEN 60
70 PRINT Q:GOSUB 5555:NO¤=K¤
80 PRINT "Kérem a férfiak csoportjait!":Q=INKEY¤:V¤="":CS=1:F¤(CS)="":IS=0:PRINT
 CHR¤(95);
90 IDO=IDO+1:IF IDO>C*MI THEN 4000
100 Q=INKEY¤:IF Q="" THEN 90
110 IF ASC(Q)=13 THEN 150
120 IF (Q<"A" OR Q>"Z") AND Q<>" " THEN PRINT TAB(46);"Hibás Karakter: ";Q:GOTO
80
125 PRINT CHR¤(8)+Q+CHR¤(95);
130 IF Q¤=" " THEN CS=CS+1:F¤(CS)="":GOTO 90
135 FOR J=0 TO IS:IF Q=F1¤(J) THEN PRINT:PRINT "A '";Q;"' Karakter kétszer szere
pel!":GOTO 80
140 NEXT J:F¤(CS)=F¤(CS)+Q:IS=IS+1:F1¤(IS)=Q:GOTO 90
150 I=0:PRINT CHR¤(8)+" ";TAB(60);"OK."
160 I=I+1
165 IF F¤(I)="" THEN CS=CS-1:FOR J=I TO CS:F¤(J)=F¤(J+1):NEXT J:IF I<CS THEN 165
170 H(I)=LEN(F¤(I)):IF I<CS THEN 160
175 IF F¤(CS)="" THEN CS=CS-1
180 IF CS=0 THEN 80
230 FOR I=1 TO CS:N¤(I)=""
240 FOR J=1 TO H(I):N¤(I)=N¤(I)+MID¤(NO¤,ASC(MID¤(F¤(I),J,1))-64,1):NEXT J
250 S¤=N¤(I):GOSUB 5555:N¤(I)=K¤:NEXT I
260 FOR I=1 TO 20:L1=RND(CS):L2=RND(CS)
270 IF H(L1)=H(L2) THEN N¤=N¤(L1):N¤(L1)=N¤(L2):N¤(L2)=N¤
280 NEXT I:PRINT "A nök csoportjai:"
290 FOR I=1 TO CS:PRINT N¤(I)+" ";:NEXT I
300 PRINT:PRINT "Akarsz tippelni? ";:Q=INKEY¤
310 IDO=IDO+1:IF IDO>C*MI THEN 4000
320 Q=INKEY¤:IF Q="" THEN 310
330 PRINT Q:IF Q="N" THEN PR=PR+1:GOTO 80
990 '
1000 '############### TIPPELÉS ##################
1010 PRINT:PRINT "A feleségek betüjele alá írd a férjekét!"
1020 PRINT NJ¤:ZZ=0:Q=INKEY¤
1030 IDO=IDO+1:IF IDO>MI THEN 5000
1040 Q=INKEY¤:IF Q="" THEN 1030
1050 IF Q<"A" OR Q>"Z" THEN PRINT:PRINT "Hibás Karakter: ";Q:PO=4:GOTO 1010
1060 ZZ=ZZ+1:PRINT Q;:QQ=MID¤(NO¤,ASC(Q)-64,1)
1070 IF QQ=MID¤(NJ¤,ZZ,1) THEN PO=PO+6
```

```
1080 IF ZZ=26 THEN 5020 ELSE 1030
3990 '
4000 '####### KÖTELEZŐ TIPPELÉS ########
4010 PRINT:PRINT "Már csak az idő";(1-C)*100;"%-a van hátra, így tippelned kell.
":GOTO 1000
4990 '
5000 '############# VÉGE ##############
5010 PRINT:PRINT:PRINT "Sajnos lejárt az időd!":PRINT
5020 PO=PO+(PR>1)*(PR-1)*20:PO=-INT(PO)*(PO>0)
5030 PRINT:PRINT:PRINT "Pontjaid száma:";PO
5040 PRINT:PRINT:PRINT:END
5554 '
5555 '######## STRING KEVERŐ RUTIN #########
5560 N=LEN(S):FOR H=1 TO N:SE(H)=MIDɑ(S,H,1):NEXT H
5570 FOR H=1 TO 5:L1=RND(N):L2=RND(N)
5580 SE=SE(L1):SE(L1)=SE(L2):SE(L2)=SE:NEXT H
5590 PK=RND(N)
5600 P=RND(3)*6+RND(2)*2-3:IF N=P*INT(N/P) THEN 5600
5610 Kɑ="":FOR H=1 TO N:PK=PK+P
5620 IF PK>N THEN PK=PK-N:GOTO 5620
5630 Kɑ=Kɑ+SE(PK):NEXT H:RETURN
```

Megjegyzés
Azoknál a HT1080Z gépre készült programoknál, amelyekben ékzetes betűk is szerepelnek, kénytelenek voltunk az ún. svéd karakterkészletet használni a nyomtatás során. Ezekben a programokban tehát dollárjel helyett ¤, és mint ahogy az lenni szokott, a @ jel helyett É szerepel.

```
5 REM #### "♥¬\LFUL ES A ♥♦/\TAK" ######
10 DIM H(26),F$(26),N$(26),SE$(52)
20 MI=32000:C=.8:PO=4
30 PRINT "□"+CHR$(14):PRINT "     ♥¬\LFULL ES A ♥♦/\TAK - JATEK.":PRINT:PRINT
40 POKE 198,0:PRINT "_EMELEM KESZEN ALLSZ (ULSZ) A JATEKRA?":PRINT:PRINT
50 FOR I=1 TO 26:S$=S$+CHR$(192+I):NJ$=NJ$+CHR$(192+I):NEXT I
60 GOSUB 5555:NO$=K$
70 ID=ID+1:GET O$:IF O$="" THEN 70
80 PRINT "/EREM A FERFIAK CSOPORTJAIT!":POKE 198,0:V$="":CS=1:F$(CS)=""
90 ID=ID+1:IF ID>C*MI THEN 4000
100 GET O$:IF O$="" THEN 90
110 IF ASC(O$)=13 THEN 150
120 IF (O$<"A" OR O$>"Z") AND O$<>" " THEN PRINT "  IIBAS KARAKTER: ";O$:GOTO 80
125 PRINT O$;
130 IF O$=" " THEN CS=CS+1:F$(CS)="":GOTO 90
135 FOR J1=1 TO CS:FOR J2=1 TO  LEN(F$(J1)):IF O$=MID$(F$(J1),J2,1) THEN 145
140 NEXTJ2,J1:F$(CS)=F$(CS)+O$:GOTO 90
145 PRINT:PRINT"♠ "+O$+" /♠_♠/I¯_ /¯I♥♦¯_ ♥♦¯_¯TL!":GOTO 80
150 I=0:PRINT:PRINT "¯GY PILLANAT!"
160 I=I+1
165 IF F$(I)="" THEN CS=CS-1:FOR J=I TO CS:F$(J)=F$(J+1):NEXT J:IF I<CS THEN 165
170 H(I)=LEN(F$(I)):IF I<CS THEN 160
175 IF F$(CS)="" THEN CS=CS-1
180 IF CS=0 THEN 80
230 FOR I=1 TO CS:N$(I)=""
240 FOR J=1 TO H(I):N$(I)=N$(I)+MID$(NO$,ASC(MID$(F$(I),J,1))-64,1):NEXT J
250 S$=N$(I):GOSUB 5555:N$(I)=K$:NEXT I
260 FOR I=1 TO 20:L1=1+INT(RND(1)*CS):L2=1+INT(RND(1)*CS)
270 IF H(L1)=H(L2) THEN NN$=N$(L1):N$(L1)=N$(L2):N$(L2)=NN$
[illegible] NEXT I:PRINT "□♠ NOK CSOPORTJAI:"
[illegible] FOR I=1 TO CS:PRINT N$(I)+" ";:NEXT I
[illegible] PRINT:PRINT "♠AKARSZ TIPPELNI? ";:POKE 198,0
[illegible] ID=ID+1:IF ID>C*MI THEN 4000
[illegible] GET O$:IF O$="" THEN 310
[illegible] PRINT O$:IF O$="N" THEN PR=PR+1:GOTO 80
```

```
990 REM
1000 REM ######### TIPPELES ###########
1010 PRINT:PRINT "♠ FELESEGEK BETUJELE ALA IRD A FERJEKET!";
1020 PRINT NJ$:POKE 198,0:SZ=0
1030 ID=ID+1:IF ID>MI THEN 5000
1040 GET Q$:IF Q$="" THEN 1030
1050 IF Q$<"A" OR Q$>"Z" THEN PRINT:PRINT " HIBAS KARAKTER: ";Q$:PO=4:GOTO 1010
1060 SZ=SZ+1:PRINT Q$;:QQ$=MID$(NO$,ASC(Q$)-64,1)
1070 IF QQ$=MID$(NJ$,SZ,1) THEN PO=PO+6
1080 IF SZ=26 THEN 5020
1090 GOTO 1030
4000 REM ###### KOTELEZO TIPPELES ######
4010 PRINT:PRINT " AR CSAK AZ IDO";(1-C)*100;"%-A VAN HATRA."
4020 PRINT "IGY TIPPELNED KELL."
4030 GOTO 1000
5000 REM ############ VEGE #############
5010 PRINT:PRINT:PRINT "♥AJNOS LEJART AZ IDOD!":PRINT
5020 PO=PO+(PR>1)*(PR-1)*20:PO=-INT(PO)*(PO>0)
5030 PRINT:PRINT:PRINT "PONTJAID SZAMA:";PO
5040 PRINT:PRINT:END
5554 REM
5555 REM ##### STRING KEVERO RUTIN #####
5560 N=LEN(S$):FOR H=1 TO N:SE$(H)=MID$(S$,H,1):NEXT H
5570 FOR H=1 TO 5:L1=1+INT(RND(1)*N):L2=1+INT(RND(1)*N)
5580 SF$=SE$(L1):SE$(L1)=SE$(L2):SE$(L2)=SF$:NEXT H
5590 PK=1+INT(RND(1)*N)
5600 P=INT(RND(1)*3)*6+5-2*(RND(1)<.5):IF N=P*INT(N/P) THEN 5600
5610 K$="":FOR H=1 TO N:PK=PK+P
5620 IF PK>N THEN PK=PK-N:GOTO 5620
5630 K$=K$+SE$(PK):NEXT H:RETURN
```

```
  5 REM #### SCILful es a SZKI-tak       ####
 10 DIM H(26): DIM F$(26,26): DIM N$(26,26)
 20 LET MI=19000: LET C=.8: LET PO=4: LET PR=0
 30 CLS : PRINT : PRINT "SCILful es a SZKItak - jatek": PRINT :
 PRINT
 40 LET Q$=INKEY$: PRINT "Remelem keszen allsz (ulsz)      a jat
ekra? ";
 50 LET S$="": FOR I=1 TO 26: LET S$=S$+CHR$ (64+I): NEXT  I: L
ET M$=S$
 60 GO SUB 5555: LET O$=K$: LET ID=0
 70 LET ID=ID+1: LET Q$=INKEY$: IF Q$="" OR Q$="n" THEN  GO TO
70
 74 PRINT Q$: PRINT
 75 IF "A" =Q$ AND Q$<="Z" THEN  PRINT : PRINT "ELOSZOR VALTS A
T KISBETURE,      UTANA INDITS UJRA!": STOP
 76 FOR I=1 TO 50: NEXT I
 80 PRINT "Kerem a ferjek csoportjait!": LET Q$=INKEY$: LET V$
="": LET CS=1: LET H(1)=0: LET F$(1)=""
 90 LET ID=ID+1: IF ID>C*MI THEN  GO TO 4000
100 LET Q$=INKEY$: IF Q$="" THEN  GO TO 90
110 IF CODE Q$=13 THEN  GO TO 150
120 IF (Q$<"a" OR Q$>"z") AND Q$<>" " THEN  PRINT : PRINT "Hiba
s karakter: ";Q$: GO TO 80
130 IF Q$=" " AND F$(CS,1)<>" " THEN  LET CS=CS+1: LET H(CS)=0:
 LET F$(CS)="": PRINT " ";: GO TO 70
132 IF Q$=" " THEN  GO TO 70
134 PRINT Q$;
135 FOR I=1 TO CS: FOR J=1 TO H(I): IF Q$=F$(I,J) THEN  GO TO 1
45
140 NEXT J: NEXT I: FOR I=1 TO (3-CS)*20: NEXT I: LET H(CS)=H(C
S)+1: LET F$(CS,H(CS))=Q$: GO TO 90
145 PRINT : PRINT "A ";Q$;" betu ketszer szerepel!": GO TO 80
150 PRINT "."
```

```
175 IF F$(CS,1)=" " THEN LET CS=CS-1
180 IF CS=0 THEN GO TO 80
230 FOR I=1 TO CS: LET N$(I)=""
240 FOR J=1 TO H(I): LET N$(I,J)=O$(CODE F$(I,J)-96): NEXT J
250 LET S$=N$(I,1 TO H(I)): GO SUB 5555: LET N$(I)=K$: NEXT I
260 FOR I=1 TO 20: LET L1=1+INT (RND*CS): LET L2=1+INT (RND*CS)
270 IF H(L1)=H(L2) THEN LET W$=N$(L1): LET N$(L1)=N$(L2): LET
N$(L2)=W$
280 NEXT I: PRINT "A nok csoportjai:"
290 FOR I=1 TO CS: PRINT N$(I,1 TO H(I))+" ";: NEXT I
300 PRINT : PRINT "Akarsz tippelni? ";: LET Q$=INKEY$
310 LET ID=ID+1: IF ID>C*MI THEN GO TO 4000
320 LET Q$=INKEY$: IF Q$="" THEN GO TO 310
330 PRINT Q$: FOR I=1 TO 50: NEXT I: IF Q$="n" THEN LET PR=PR+
1: GO TO 80
990 REM
1000 REM ##### TIPPELES ########
1010 PRINT : PRINT "A feleseqek betujele ala ird   a ferjeket!"
1020 PRINT M$: LET Q$=INKEY$: LET SZ=0
1030 LET ID=ID+1: IF ID>MI THEN GO TO 5000
1040 LET Q$=INKEY$: IF Q$="" THEN GO TO 1030
1050 IF Q$<"a" OR Q$>"z" THEN PRINT : PRINT "Hibas karakter: ";
Q$: LET PO=4: GO TO 1010
1060 LET SZ=SZ+1: PRINT Q$;: LET W$=O$(CODE Q$-96)
1065 FOR I=1 TO 50: NEXT I
1070 IF W$=M$(SZ) THEN LET PO=PO+6
1080 IF SZ=26 THEN GO TO 5020
1090 GO TO 1030
3990 REM
4000 REM ### KOTELEZO TIPPELES #
4010 PRINT : PRINT "Mar csak az ido";(1-C)*100;"%-a van"
4020 PRINT "hatra, igy tippelned kell."
4030 GO TO 1000
4990 REM
5000 REM ######## VEGE ########
5010 PRINT : PRINT : PRINT "Sajnos lejart az idod!": PRINT
5020 LET PO=PO-(PR>1)*(PR-1)*20: LET PO=INT PO*(PO>0)
5030 PRINT : PRINT : PRINT "Pontjaid szama:";PO
5040 PRINT : PRINT : STOP
5554 REM
5555 REM #### STRING KEVERO ###
5560 LET N=LEN S$
5570 FOR H=1 TO 5: LET L1=1+INT (RND*N): LET L2=1+INT (RND*N)
5580 LET W$=S$(L1): LET S$(L1)=S$(L2): LET S$(L2)=W$: NEXT H
5590 LET PK=1+INT (RND*N)
5600 LET P=INT (RND*3)*6+5+2*(RND<.5): IF N=P*INT (N/P) THEN GO
 TO 5600
5610 LET K$="": FOR H=1 TO N: LET PK=PK+P
5620 IF PK>N THEN LET PK=PK-N: GO TO 5620
5630 LET K$=K$+S$(PK): NEXT H: RETURN
```

A program felépítése, működése

A program az induláskor egy stringet feltölt 26 nagybetűvel ábécésorrendben (a HT1080Z-ben mindig fordítva, tehát ahol nagybetűt írunk, ott értelemszerűen kisbetű értendő és viszont). Ezt a stringet megőrzi az NJ$-ban (Sinclair Spectrumon M$), és meghívja a keverőrutint, amely összekeveri ezt a betűcsoportot, s ezt tárolja a NO$-ban (Sinclair Spectrumon O$-ban). Tulajdonképpen ez lesz a kitalálandó párosítás. Pl. NO$ harmadik karaktere jelenti, hogy ki lesz „c" férfi felesége. A stringkeverő rutin 5555-nél kezdődik. Ez lényegében megegyezik a vetélkedő selejtezőjének egyik „elrontott program" feladatával (l. a 31...39. oldalon). A rutin

S$-ban kapja meg a keverendő stringet, és K$-ban adja vissza a kevertet. (Ezt a rutint kapták meg a versenyzők a döntőben.)

A program méri az időt, s az idő bizonyos részének letelte után tippelésre kötelezi a játékost. Ez a művelet a 4000-nél kezdődő részben történik. Azt, hogy mennyi az a bizonyos rész, amelynél ez a tippelési felszólítás megtörténik, azt a C változóban adjuk meg. Ha a játékos teljes ideje lejárt, a program az 5000-nél kezdődő „vége" részre megy, ahol elvégzi a pontszámítást, és ki is írja a végső pontértéket a képernyőre.

A program a megfelelő értékadások és a kezdő feliratkozások után, ha a versenyző megadja a jelt, a férfiak csoportjainak beírásával indul (80-as sor).

Minden egyes karakter beírásakor ellenőrzi a gép, hogy az kisbetű, betűköz vagy RETURN-e; ha egyik sem, azonnal hibát jelez. Ellenőrzi azt is, hogy az utoljára beütött betű nem szerepelt-e már egyszer; ha igen, ezt is azonnal kijelzi és újra kérdez.

A beírás után következik a feldolgozás (150-es sor). A beírt csoportok számát a CS változóban, a csoportokat az F$ tömbben tároljuk. Tekintettel arra, hogy a játékos dolgának megkönnyítése érdekében egymás után több betűközt — space-t — is elfogad a gép, a feldolgozást az ún. üres csoportok kiszórásával kell kezdeni. (A több egymás utáni betűköz ilyen üres csoportként jelentkezik a gép számára.)

Ezután következik a kódolás, azaz a megfelelő férfi csoportokhoz a nők stringjében tárolt betűk kiválasztásával megkeresi a program a megfelelőket (a „feleségeket"). Ezeket a csoportokat az N$ vektorban tárolja. Ám egy csoport összeállítása után a csoportba tartozó nők betűit még meg is kell keverni, hogy ne a férfiak sorrendjének megfelelő sorrendben írja ki a képernyőre! Minden egyes csoport összeállítása után behívjuk tehát a keverőrutint, és megkeverjük a csoportot.

A kevert stringet azután mindig visszaküldjük az N$ vektor megfelelő elemébe. Ha programunk végzett valamennyi csoporttal, még meg kell „kutyulnia" az azonos létszámú csoportokat. Ezt egyszerűen úgy végzi el, hogy hússzor egymás után kijelöl véletlenszerűen két-két csoportot, s ha a két kijelölt csoport létszáma azonos, akkor egyszerűen megcseréli azokat. (Ezt a lépést sok versenyzőnk kifelejtette a programjából, pedig fontos!) Ezután már nincs más hátra, mint visszaírni a nők megfelelő csoportjait a képernyőre, majd megkérdezni, hogy akar-e a játékos tippelni. Ha igen, úgy elmegy a program a tippelőrészre (1000-es sortól). Ha viszont nem, úgy programunk megnöveli a próbálkozások számát 1-gyel, s ismét kérdezi a férfiak csoportjait.

A tippelőrész tehát az 1000-es sornál kezdődik. Itt a program kiírja a kérdés után az NJ$-t, azaz a nőket ábécérendben. Ezek alá kell beírni a megfelelő férfiakat. Idegen karakter esetén azonnal jelzi a program, hogy rossz karaktert írtunk be, és ezért újrakérdez. Jó karakter esetén az 1060-as sorban megállapítja, hogy ki az általunk megadott férfinak a felesége, s ha ez megegyezik az általunk beírt „férfibetű" felett álló betűvel — azaz helyes a válaszunk —, akkor megnöveli a pontok számát 6-tal. Ha mind a 26 férjet beírtuk, úgy a program a „végerész" közepére megy, átugorja a „lejárt az idő" feliratot, és végül kiírja a végeredményt.

Különbségek a négy gép programjában

A fő különbségeket a különböző karakterkódok, az INKEY$ és a GET működésének különbözőségei, a gépi idők mérésének különbségei és persze a Sinclair Spectrumnak a stringtömbökkel való másféle bánásmódja okozza. A HT1080Z-nél és a Sinclair Spectrumnál az ismétlődő karakter kiszűrése miatti vizsgálat a sokadik férfi beírásánál eléggé lelassítja a programot, így nem lehet gyors egymásutánban beírni a karaktereket. Épp ezért itt szükségessé vált a betűköz leütésének érzékelése is. Ezt a HT1080Z esetén a kurzor megjelenésével, a Sinclair Spectrum esetén aláhúzásjellel oldottuk meg. A Sinclair Spectrumnál azt is meg kellett oldani, hogy az egyszer lenyomott billentyűt ne vegye be egyből három egymásutáninak, ezért ide várakozó

ciklusokat építettünk be. A RETURN gomb lenyomása után a gép egy ideig gondolkodik, így a játékossal érzékeltetni kell valamilyen módon, hogy a gép elfogadta a RETURN-t. Erre különféle megoldásokat találtunk.

További különbség, hogy az M08X az egyetlen gép a négy közül, amely ismeri a SWAP utasítást, ami az utána következő két változó értékét megcseréli. Ezt a többi gépen hagyományos módszerrel kellett megoldani.

E program négy változata ismét érdekes összevetésekre ad lehetőséget.

Tessék, tegyék meg kedves olvasóink!

Akárcsak a vetélkedő selejtezőjében, a döntőben is volt olyan, rövid idő alatt végrehajtandó feladatsor, amelyet tesztnek neveztünk. A nagy különbség a selejtező és a döntő tesztje között az volt, hogy utóbbit a számítógép mellett kellett megoldani, s a beírt válaszokat a gép azonnal ki is értékelte. Tehát a 15 percnyi megoldási idő végén a gép mindenkivel közölte a pontszámát. (A papír és a ceruza használatát engedélyeztük a megoldásokhoz.)

Igyekeztünk valamennyi tesztfeladatot átírni a másik három gépre, de erre nem nagyon volt szükség, hiszen a kérdések többsége nem számítógépes kérdés. Így hát a következőkben bemutatjuk a teszt kérdéseit (ahol szükséges, gépenként).

```
 1. SZAMU FELADAT:

Ket vadasz all egymastol 16 km tavolsagra, az egyiket Lewis Sinclairnek,
a masikat Hewlett Packardnak hivjak. Egyszerre elindulnak egymas fele,
mindketten egyenletes, 4 km/oras sebesseggel mennek. Ugyanekkor Sinclair
vizslaja, Spectrum, elkezd loholni Packard fele. Mikor odaer, azonnal
megfordul, s megy vissza a gazdajahoz, ahol ujra megfordul, szalad Packardhoz,
s igy tovabb, mig a ket ember talalkozik. Tudjuk, hogy Spectrum is egyenletes,
de 16 km/oras sebesseggel fut.


Ezekutan az a kerdes, hogy hany km-t tesz meg a kutya addig a pillanatig,
mig a ket vadasz talalkozik?          (A valasz utan nyomd meg a CR gombot !)


 2. SZAMU FELADAT:

#######
 #######
  #######
   #######
  #######
 #######
#######

Melyik programsor allitja elo ezt az abrat az alabbiak kozul?

1 A$=STRING$(7,35):FOR I=1 TO 7:PRINT TAB 4-ABS(4-I));A$:NEXT I
2 B$=CHR$(35):A$=B$+B$+B$:A$=A$+A$+B$:FOR I=1 TO 4:PRINT TAB(I);A$:NEXT I:FOR
I=4 T[illegible] STEP -1:PRINT TAB I;A$:NEXT I
3 B$=CHR$(35):A$=STRING$(7,B$):FOR I=0 TO 3:PRINT TAB(I+1);A$:NEXT I:FOR I=2 TO
0 STEP -1:PRINT TAB(I+1);A$:NEXT I
```

```
1 A$="#######":FOR I=1 TO 7:PRINT TAB(4-ABS(4-I));A$:NEXT I
2 B$="#":A$=B$+B$:A$=A$+A$+A$+B$:FOR I=1 TO 4:PRINT TAB(I);A$:NEXT I:FOR I=4 TO 1 STEP -1:PRINT TAB(I);A$:NEXT I
3 B$="###":A$=B$+B$+"#":FOR I=0 TO 3:PRINT TAB(I+1);A$:NEXT I:FOR I=2 TO 0 STEP -1:PRINT TAB(I+1),A$:NEXT I
```

```
1 LET A$="#######": FOR I=1 TO 7: PRINT TAB (4-ABS (4-I));A$: NEXT I
2 LET B$="#": LET A$=B$+B$: LET A$=A$+A$+A$+B$: FOR I=1 TO 4: PRINT TAB I;A$: NEXT I: FOR I=4 TO 1 STEP -1: PRINT TAB I;A$: NEXT I
3 LET B$="###": LET A$=B$+B$+"#": FOR I=0 TO 3: PRINT TAB (I+1);A$: NEXT I: FOR I=2 TO 0 STEP -1: PRINT TAB (I+1),A$: NEXT I
```

---

a) Az 1 es a 2 sorszamu   b) Csak az 1   c) Csak a 3
d) Az 1 es a 3   e) A 2 es a 3

3. SZAMU FELADAT:

Hany darab space van a kovetkezo sorban az aposztrofok kozott?

```
10 REM '
```

a) 48   b) 50   c) 52   d) 54   e) egyeb

4. SZAMU FELADAT:

Mit szamol a kovetkezo program?

```
10 FOR I=0 TO 2
20 INPUT X(I),Y(I)
30 NEXT I
40 FOR J=0 TO 2
50 XX=XX+X(J):YY=YY+Y(J)
60 NEXT J
70 XX=XX/3:YY=YY/3
80 FOR J=0 TO 2
90 E(J)=SQR((X(J)-XX)*(X(J)-XX)+(Y(J)-YY)*(Y(J)-Y
100 PRINT 1.5*E(J),
110 NEXT J
120 END
```

Megjegyzés
A Sinclair Spectrum gépen természetesen LET utasítások szükségesek.

a) Egy csucsaival adott haromszog sulypontjat   b) 3 oldalat
c) 3 sulyvonalat   d) 3 magassagat   e) 3 szogfelezojet

5. SZAMU FELADAT:

Mi tortenik a kovetkezo programreszlet futasa soran, ha PR erteke 15 ?

```
100 ON 1-(PR<11)-(PR<13)-(PR<17) GOSUB 200,300,400,500
200 PRINT A$:RETURN
300 PRINT B$:RETURN
400 PRINT C$:RETURN
500 PRINT D$:RETURN
600 END
```

```
100 GO SUB 200+((PR<11)+(PR<13)+(PR<17))*100
200 PRINT A$: RETURN
300 PRINT B$: RETURN
400 PRINT C$: RETURN
500 PRINT D$: RETURN
600 STOP
```

---

a) Kiirja egymas utan a B$,A$,B$,C$ es D$-t   b) Kiirja B$-t
c) Kiirja C$-t   d) Kiirja B$-t,A$-t, majd hibajelzessel megall   e)mast

6. SZAMU FELADAT:

Hanyfelekeppen olvashato ki az alabbi abrabol a BUDAPEST szo ?

```
B U D A P
U D A P E
D A P E S
A P E S T
```

a) 20   b) 8*7*6*5*4*3*2*1   c) 35   d) 128   e) 30

Megoldás

Az 1. feladatra a helyes válasz: 32
A 2. feladatra a helyes válasz: b
A 3. feladatra a helyes válasz: b
A 4. feladatra a helyes válasz: c
Az 5. feladatra a helyes válasz: d
A 6. feladatra a helyes válasz: c

A döntőbe jutottak, mint azt már a döntő bevezetőjében írtuk, kaptak egy házi feladatot. Olyan programot kellett írniuk, amely tartalmában az SzKI—SCI-L—SCITEL intézetek, ill. gépeik, az M08X—PROPER-8—PROPER-16 reklámja, programozási tartalmát illetően pedig képernyőkezelési feladat. Ehhez a programhoz segítségképpen minden versenyző megkapta postán az M08X gép BASIC-leírását, valamint az alábbi kis kiegészítést. Az otthon megírt programot a döntő előtt, ill. után minden versenyző gépbe írhatta, s ennek során segítségképpen használhatta az általunk készen adott kurzormozgató rutint.

A következő kis magyarázatot, amely tehát a versenyzőknek kiadott segédlettel megegyezik, azért tesszük közzé könyvünkben, mert ez talán az M08X felhasználói (programozói) számára is hasznos lehet.

Speciális képernyőkezelés az M08X számítógépen

Az M08X a képernyőt a legegyszerűbb módon kezeli, azaz csak „ROLL" módusban működik a képernyő. Ennek következtében a kurzor csak egyesével jobbra léphet, ill. CR esetén a következő sor elejére.

A következőkben megadjuk a kurzor mozgatásának módját, valamint példát adunk olyan képernyőkezelésre, ami nem az aktuális kurzorpozícióban hajtja végre az egyes író/olvasó műveleteket. Ez utóbbi a képernyőkezelő programcsomagok nélkülözhetetlen része.

Az M08X képernyőjének időről időre „felfrissítésre" van szüksége; ez azt jelenti, hogy annak tartalmát ismételten meg kell jeleníteni, mielőtt a képernyő elhalványulna. Ezt az ismételt megjelenítést a rendszer végzi, a felhasználónak semmit sem kell tennie, hogy ez végbemenjen. Fontos azonban az a tárcím, ahonnan az előbb leírt „felfrissítés" kezdődik. Ami ui. a felfrissítési tartományon belül van, az mind megjelenik a képernyőn, és nem szükséges semmilyen kiviteli (OUT) műveletet végrehajtani. A képernyő-felfrissítési tár kezdőcíme 63536, mert a 64 Kbyte kapacitású tár utolsó 2000 byte-ja van fenntartva a 80×25 karakter nagyságú képernyő felfrissítésére. Például a POKE 63536,65 a képernyő bal felső, a POKE 65535,65 a jobb alsó sarkába helyez egy A betűt.

Az M08X gép képernyőjén az adatokat bizonyos attributumokkal lehet megjeleníteni: ezek az „aláhúzott", „fordított", „villogó" vagy „világos" megjelenítési módok a „normál" — tehát sötét alapon világos — mellett. Ezeket az attributumokat a képernyő felfrissítési területén elhelyezett attributumbyte-ok elhelyezésével állíthatjuk be. Az attributumbyte felépítése a következő:

10UR00BH, ahol az

U = 1 „aláhúzott"
R = 1 „fordított"
B = 1 „villogó"
H = 1 „világos"

attributumot jelent!

Az attributumbyte hatása arra az adatmezőre érvényes, ami a felfrissítési területen tőle jobbra áll (tehát a képernyőn jobbra, ill. alatta) és egy újabb attributumbyte megjelenéséig tart, vagy a felfrissítési tartomány végéig (a képernyő jobb alsó pozíciójáig).

Az egyes attributumokat egymással kombinálhatjuk. Ha minden attributumbit az attributumbyte-ban = 0, akkor a „normál" attributum érvényes. Ekkor az attributumbyte értéke 128.

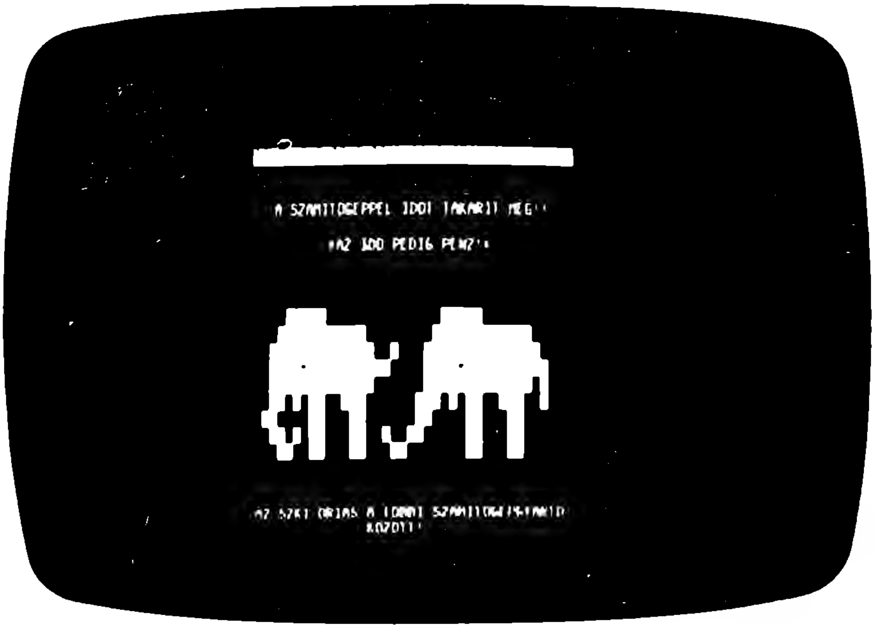
További megjegyezések:

1. Vigyázat! Egy „világos" attributumot nem lehet PRINT CHR$ (129)-cel elhelyezni a képernyőre, csak POKE-kal. (Vagy egy ennél lényegesen bonyolultabb módszerrel, aminek ismertetésére nem térünk ki.)
2. Ne feledjük el, hogy az attributum a képernyőnek megfelelő tárterületen egy byte-ot foglal el, s így a képernyő megfelelő helyére nem kerülhet semmilyen karakter!
3. A teljes képernyő törlését PRINT CHR$ (17)-tel lehet elvégeztetni.
4. A PRINT utasítás általában mindig a legalsó sorba ír ki (ennek megkerüléséről még alább lesz szó). A vessző, a pontosvessző és a TAB használható, de egy sorban visszafelé tabulálni nem lehet.
5. Ha az utolsó sorba írt dolog után nem áll vessző vagy pontosvessző, akkor ez automatikusan SCROLL-t fog eredményezni, minden sor eggyel feljebb lép (a megfelelő attributumbyte-ok is!), az első sor pedig „elvész".
6. Vigyázat! A pontosvesszővel lezárt PRINT után, ha a következő PRINT utasítás első tétele már nem fér ki abba a sorba, akkor a következő sor elején kezdi! Ezt úgy oldhatjuk fel, hogy közben kiadunk egy PRINT CHR$ (27) utasítást.
7. Végül a kurzor mozgatásáról. A versenyzők rendelkezésére bocsátunk egy szubrutint, amely 9998—10060-ig tart. E szubrutin használata a következő: a SOR változóba betöltjük, hogy melyik sorba (0—24), az OSZ nevűbe pedig hogy melyik oszlopba (0—79) akarjuk mozgatni a kurzort. Ezután GOSUB 9999 hívás hatására a kurzor a megadott helyre kerül; a következő PRINT utasítás ettől a helytől kezd el kiírni. A szubrutin belső változói: S1, S2, O1, O2, AAA$.

Nos, a kis reklámprogramok közül a zsüri legnagyobb „elképedésére" néhány olyan jól sikerült, hogy valósággal levegő után kapkodtunk, amikor megláttuk. Különösen Kisdi Bálint programja hozta izgalomba az értékelőket. A zsüri egyik — vizuális művészetekkel foglalkozó — tagja azt mondta, ez olyan, mint egy képzőművészeti alkotás videóra. Úgy gondoltuk, érdemes tehát nyilvánosságra hoznunk a legjobb két reklámprogramot. Az egyik az említett Kisdi Bálint által készített program, a másik pedig Erdei András műve. Először mindkét programról képernyőfotókat közlünk, majd bemutatjuk a hozzájuk tartozó programot. Ha valaki ki akarja próbálni őket, azaz gépbe akarja írni, annak feltétlenül szüksége lesz kurzormozgató rutinjainkra, így Erdei András programjának részeként — a program végén — ezt is közöljük.

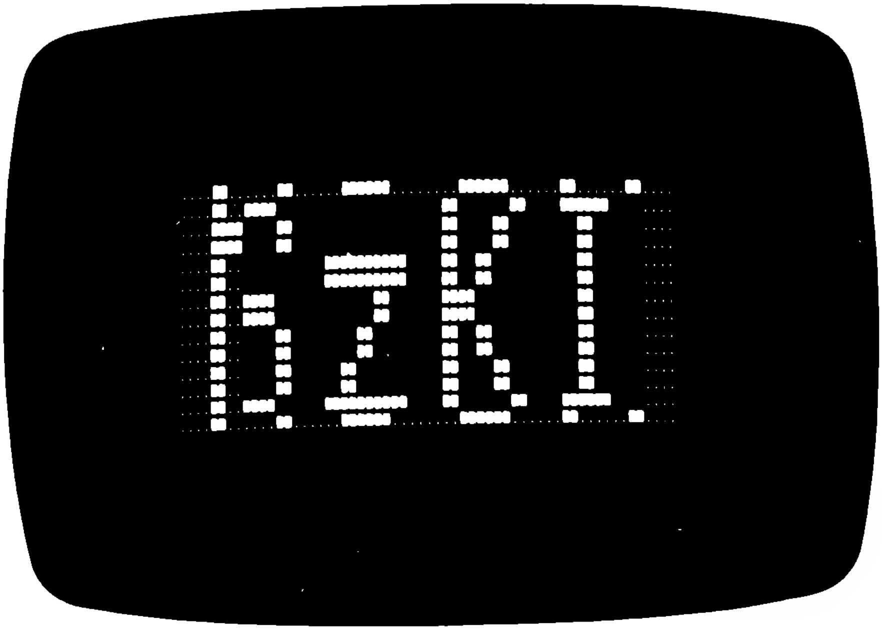
(Ezt a rutint persze az, akinek van M08X-e, egyéb dolgokra is jól használhatja.)

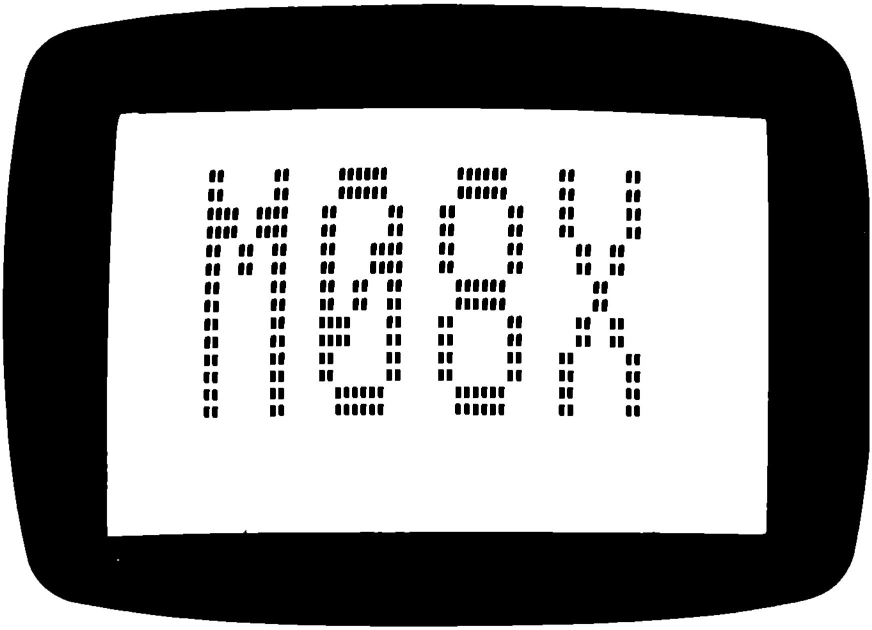
```
10 PRINT CHR$(17):GOTO 50
20 SOR=X-1:OSZ=Y-1:FOR Q=1 TO 500:GOSUB 9999:PRINT A$:A$=RIGHT$(A$,1)+LEFT$(A$,L
EN(A$)-1):IF INKEY$<>" "THEN FOR I=1 TO 50:NEXT I:NE
XT Q:RETURN ELSE RETURN
30 SOR=X-1:OSZ=Y-1:GOSUB 9999:PRINT A$: RETURN
40 X=X-1:Y=Y-1:PO=63535!+(80*X)+Y: POKE PO,A:RETURN
50 X=4:Y=24:A$="A SZAMITOGEPPEL IDOT TAKARIT MEG!!":GOSUB 30
60 X=6:Y=31:A$="*AZ IDO PEDIG PENZ!*":GOSUB 30
70 X=22:Y=22:A$="AZ SZKI ORIAS A TOBBI SZAMITOGEPGYARTO":GOSUB 30
80 X=23:Y=36:A$="KOZOTT!":GOSUB 30
90 FOR P=1 TO 75:READ X,Y,A:GOSUB 40:NEXT P
95 POKE 63535!+21*80+19,130:POKE 63535!+22*80+70,128
100 DATA 10,26,144,10,32,128,10,45,144,10,51,128
110 DATA 11,25,144,11,37,128,11,44,144,11,57,128
120 DATA 12,24,144,12,38,128,12,39,144,12,41,128,12,43,144,12,58,128
130 DATA 13,24,144,13,40,128,13,43,144,13,59,128
140 DATA 14,24,144,14,38,128,14,43,144,14,57,144,14,59,128
150 DATA 15,24,144,15,27,144,15,29,144,15,32,128,15,33,144,15,37,128,15,42,144,1
5,46,144,15,48,144,15,51,128,15,52,144,15,56,128,15,
57,144,15,59,128
160 DATA 16,23,144,16,26,128,16,29,144,16,32,128,16,34,144,16,37,128,16,42,144,1
6,45,128,16,48,144,16,51,128,16,53,144,16,56,128
170 DATA 17,24,144,17,27,144,17,29,144,17,32,128,17,34,144,17,37,128,17,38,144,1
7,40,128,17,41,144,17,44,128,17,48,144,17,51,128,17,
53,144,17,56,128
180 DATA 18,25,144,18,28,128,18,29,144,18,32,128,18,34,144,18,37,128,18,39,144,1
8,42,128,18,43,144,18,51,128,18,53,144,18,56,128
190 SOR=12:OSZ=27:GOSUB 9999:PRINT"*";:SOR=12:OSZ=46:GOSUB 9999:PRINT "*";
200 POKE 63555!,144:POKE 63596!,128:X=1:Y=21:A$="SZKI SZKI SZKI SZKI SZKI SZKI S
ZKI SZKI ":GOSUB 20
210 RESTORE : GOTO 10
8887 REM ###############################
8888 REM ##### SUBR. KURZOR TIPUS #####
8889 REM ##### TIP=A kurzor tipusa. ###
8890 REM ##### 0: Villogo, forditott ##
8891 REM ##### 1: Villogo, alahuzott ##
8892 REM ##### 2: Forditott ###########
8893 REM ##### 3: Alahuzott ###########
8894 REM ###############################
8900 TIP=INT(ABS(TIP)):IF TIP<0 OR TIP>3 THEN RETURN
8910 PRINT CHR$(27);"[";CHR$(TIP+48);"q";
8920 RETURN
9998 REM  ###############################
9999 REM  #### SUBR. KURZOR POZ. #####
10000 REM #### SOR=Sorszam (0-24) ####
10001 REM #### OSZ=Oszlopszam (0-79) #
10002 REM #### Hasznalt valtozok: ####
10003 REM #### S1, S2, O1, O2, AAA$. #
10004 REM ###########################
10010 SOR=INT(ABS(SOR)):OSZ=INT(ABS(OSZ))
10020 IF SOR<0 OR SOR>24 OR OSZ<0 OR OSZ>79 THEN RETURN
10030 S1=INT(SOR/10)+48:S2=SOR MOD 10+48:O1=INT(OSZ/10)+48:O2=OSZ MOD 10+48
10040 AAA$="["+CHR$(S1)+CHR$(S2)+CHR$(O1)+[illegible]
10045 PRINT CHR$(27
10050 PRINT CHR$(27);AAA$;
10055 PRINT CHR$(
10060 RETURN
```

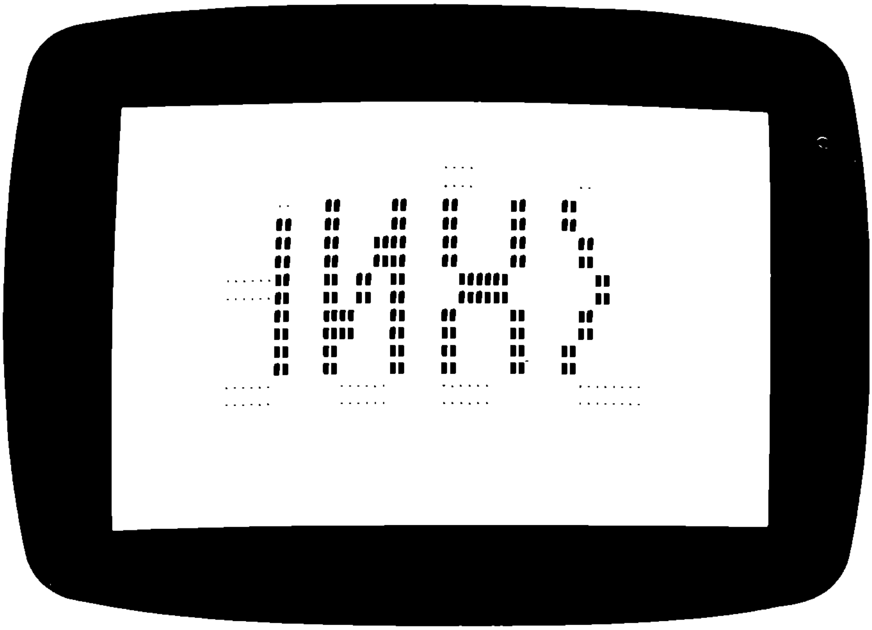


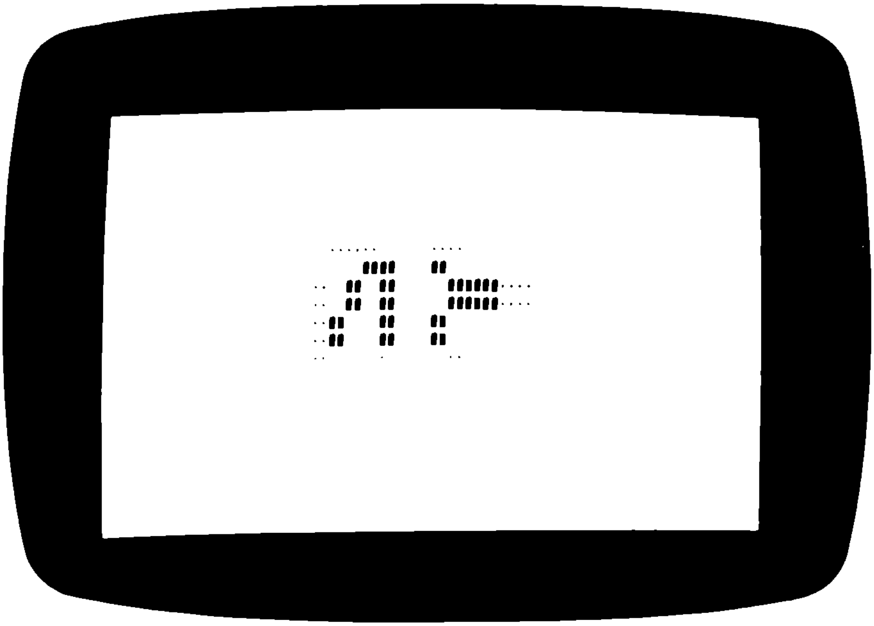
37. **képernyőkép**
Reklámprogram a képernyőn

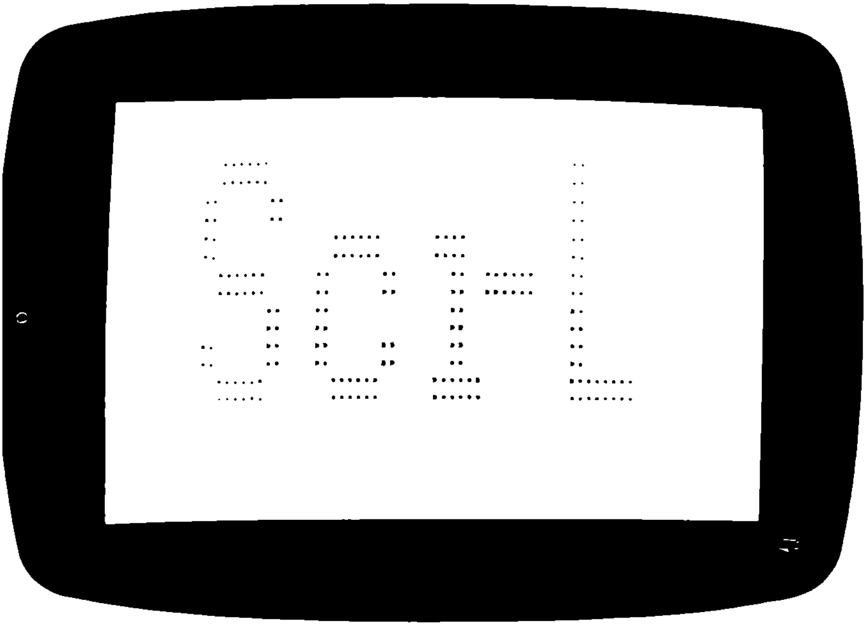
```
50 GOTO 110

60 IF F AND A(K) THEN POKE P,C:POKE P+1,C
                 ELSE POKE P,D:POKE P+1,D

70 RETURN

90 P(I)=P0+80*Y+X:A(I)=2^(Y*2-3):B(I)=X*2:I=I+1:RETURN

100 REM DEKLARACIO, INICIALIZALAS

110 DEFINT A-Z:DEFSNG P:FOR Z=1 TO 25:PRINT:NEXT Z:N=211:P0=63376!

120 DIM S(3,39),P(N),A(N),B(N),C(3),D(3),Q(3)

130 FOR I=0 TO 2:FOR J=6 TO 31:READ S(I,J):NEXT J,I:
    FOR J=6 TO 31:S(3,J)=S(1,J):NEXT J

140 DATA 38,73,73,73,50,,,68,100,84,76,68,,,127,8,20,34,65,,,65,127,65,0,0,
         127,2,4,2,127,,,62,81,73,69,62,,,54,73,73,73,54,,,99,20,8,20,99,
         38,73,73,73,50,,,56,68,68,68,40,,,69,125,64,9,3,3,,,127,64,64,64

150 P(1)=P0+1076:I=2:X1=36:X2=40:Y1=12:Y2=13

160 FOR Z=1 TO 7

170 Y=Y2:FOR X=X1+4 TO X2 STEP  4:GOSUB 90:NEXT X:X1=X1-4:
    X=X2:FOR Y=Y2-1 TO Y1 STEP -1:GOSUB 90:NEXT Y:Y2=Y2+1:
    Y=Y1:FOR X=X2-4 TO X1 STEP -4:GOSUB 90:NEXT X:X2=X2+4:
    X=X1:FOR Y=Y1+1 TO Y2          :GOSUB 90:NEXT Y:Y1=Y1-1

180 NEXT Z

190 FOR Z=0 TO 3:READ C(Z),D(Z),Q(Z):NEXT Z

200 DATA 127,32,128, 77,46,129, 42,32,144, 127,46,144

240 REM RAJZOLASOK

250 FOR J=0 TO 3:C=C(J):D=D(J):POKE 63615!,Q(J)

260 FOR K=1 TO N-1:
        P=P(K):F=S(J,B(K))       :GOSUB 60:
        P=P+2 :F=S(J,B(K)+1)     :GOSUB 60:
    NEXT K

270 NEXT J:GOTO 250
```

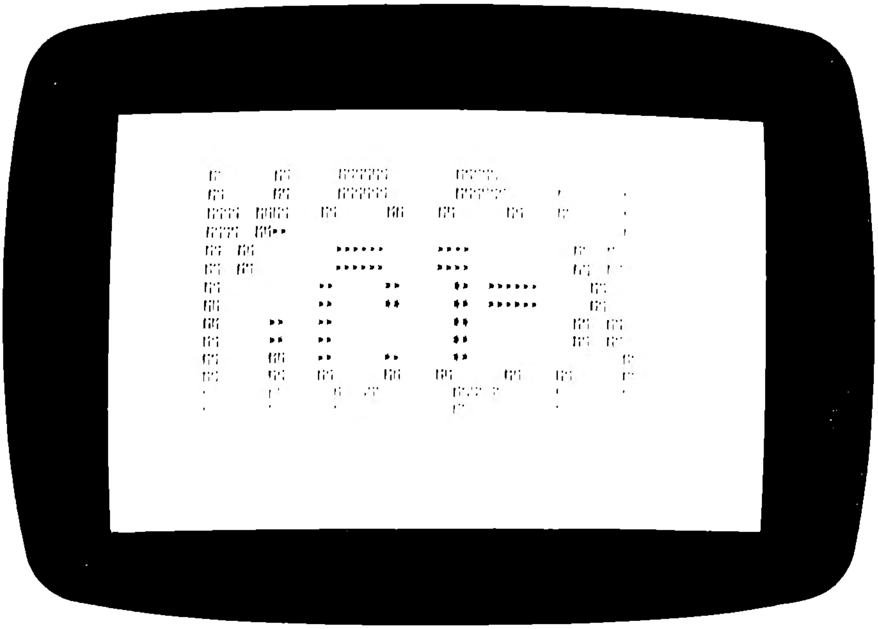


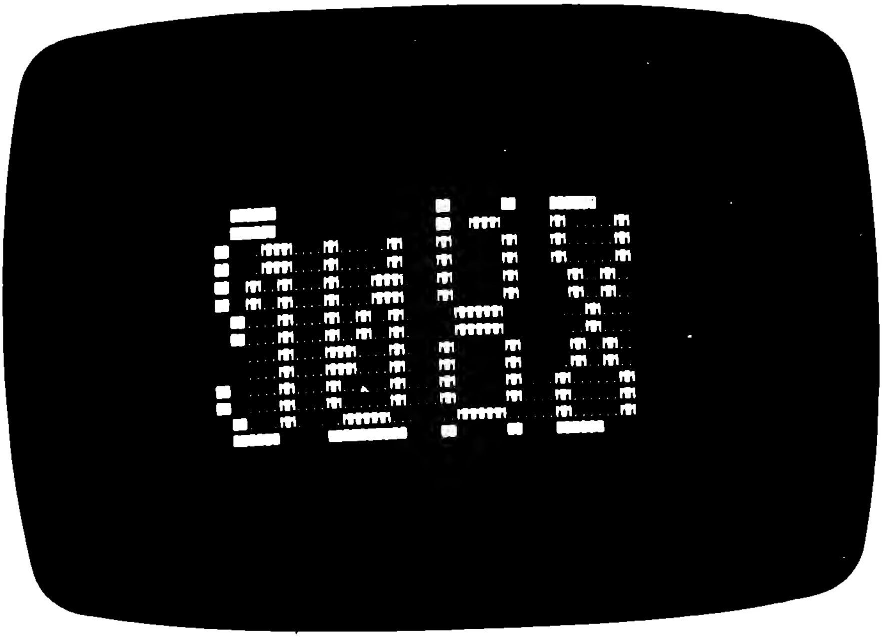


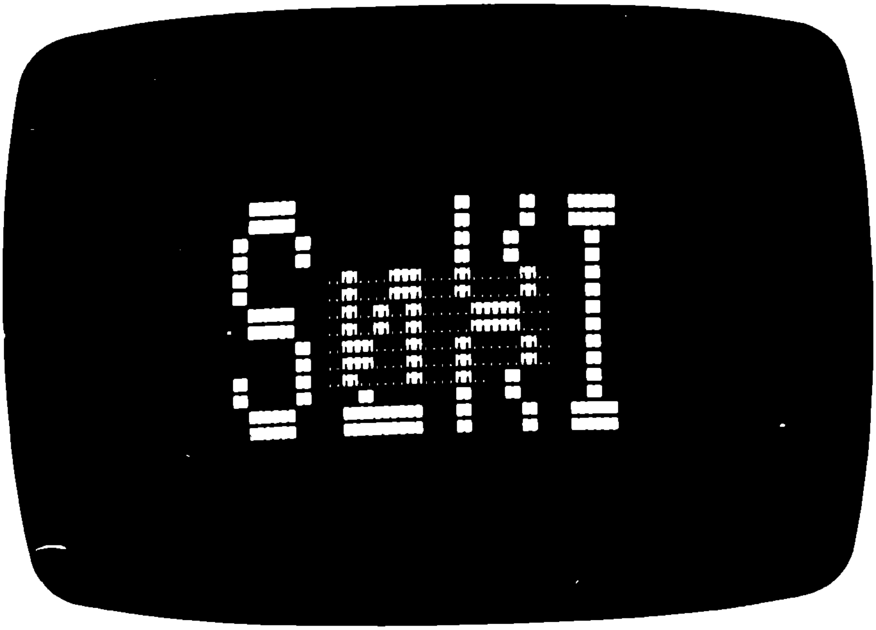


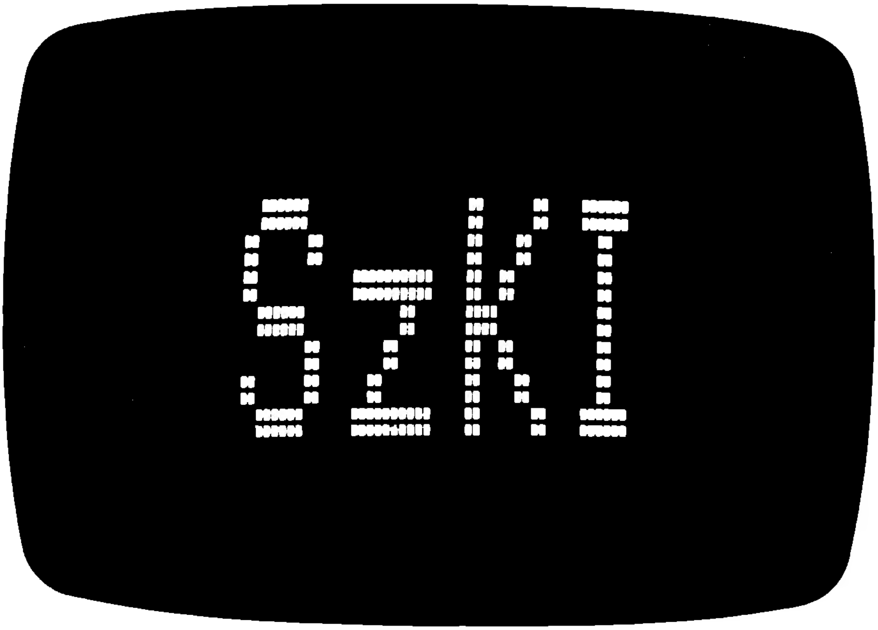


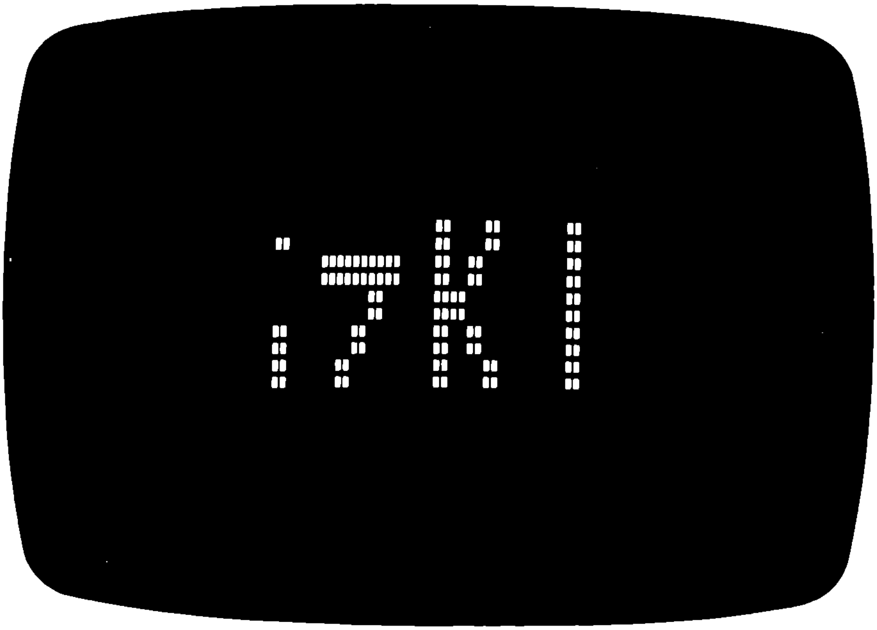

ScrL

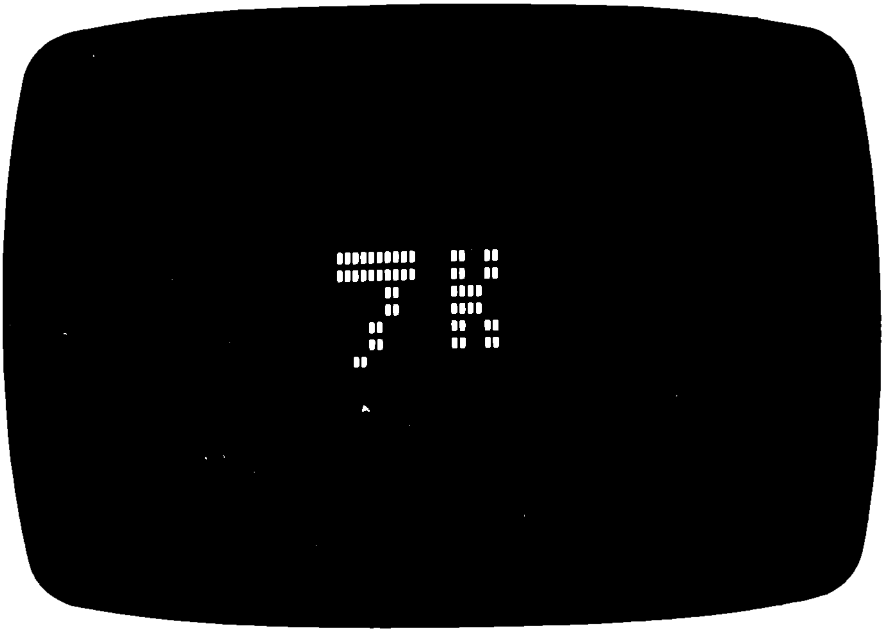







SzKI




7K